John Baillieul
Tariq Samad
Editors-in-Chief

Encyclopedia of Systems and Control

Encyclopedia of Systems and Control

John Baillieul • Tariq Samad

Editors

Encyclopedia of Systems and Control

With 416 Figures and 21 Tables

🐎 Springer Reference

Editors
John Baillieul
Mechanical Engineering
Electrical and Computer Engineering
Boston University
Boston, MA, USA

Tariq Samad
Honeywell Automation and
Control Solutions
Golden Valley, MN, USA

Printed on acid-free paper

Preface

The history of Automatic Control is both ancient and modern. If we adopt the broad view that an automatic control system is any mechanism by which an input action and output action are dynamically coupled, then the origins of this encyclopedia's subject matter may be traced back more than 2,000 years to the era of primitive time-keeping and the clepsydra water clock perfected by Ctesibius of Alexandria. In more recent history, frequently cited examples of feedback control include the automatically refilling reservoirs of flush toilets (perfected in the late nineteenth century) and the celebrated fly-ball steam-flow governor described in J.C. Maxwell's 1868 Royal Society of London paper—"On Governors."

Although it is useful to keep the technologies of antiquity in mind, the history of systems and control as covered in the pages of this encyclopedia begins in the twentieth century. The history was profoundly influenced by work of Nyqvist, Black, Bode, and others who were developing amplifier theory in response to the need to transmit wireline signals over long distances. This research provided major conceptual advances in feedback and stability that proved to be of interest in the theory of servomechanisms that was being developed at the same time. Driven by the need for fast and accurate control of weapons systems during World War II, automatic control developed quickly as a recognizable discipline.

While the developments of the first half of the twentieth century are an important backdrop for the *Encyclopedia of Systems and Control*, most of the topics directly treat developments from 1948 to the present. The year 1948 was auspicious for systems and control—and indeed for all the information sciences. Norbert Wiener's book *Cybernetics* was published by Wiley, the transistor was invented (and given its name), and Shannon's seminal paper "A Mathematical Theory of Communication" was published in the *Bell System Technical Journal*. In the years that followed, important ideas of Shannon, Wiener, Von Neumann, Turing, and many others changed the way people thought about the basic concepts of control systems. The theoretical advances have propelled industrial and societal impact as well (and vice versa). Today, advanced control is a crucial enabling technology in domains as numerous and diverse as aerospace, automotive, and marine vehicles; the process industries and manufacturing; electric power systems; homes and buildings; robotics; communication networks; economics and finance; and biology and biomedical devices.

It is this incredible broadening of the scope of the field that has motivated the editors to assemble the entries that follow. This encyclopedia aims to help students, researchers, and practitioners learn the basic elements of a vast array of topics that are now considered part of systems and control. The goal is to provide entry-level access to subject matter together with cross-references to related topics and pointers to original research and source material.

Entries in the encyclopedia are organized alphabetically by title, and extensive links to related entries are included to facilitate topical reading—these links are listed in "Cross-References" sections within entries. All cross-referenced entries are indicated by a preceding symbol: ▶. In the electronic version of the encyclopedia these entries are hyperlinked for ease of access.

The creation of the *Encyclopedia of Systems and Control* has been a major undertaking that has unfolded over a 3-year period. We owe an enormous debt to major intellectual leaders in the field who agreed to serve as topical section editors. They have ensured the value of the opus by recruiting leading experts in each of the covered topics and carefully reviewing drafts. It has been a pleasure also to work with Oliver Jackson and Andrew Spencer of Springer, who have been unfailingly accommodating and responsive over this time.

As we reflect back over the course of this project, we are reminded of how it began. Gary Balas, one of the world's experts in robust control and aerospace applications, came to one of us after a meeting with Oliver at the Springer booth at a conference and suggested this encyclopedia—but was adamant that he wasn't the right person to lead it. The two of us took the initiative (ultimately getting Gary to agree to be the section editor for the aerospace control entries). Gary died last year after a courageous fight with cancer. Our sense of accomplishment is infused with sadness at the loss of a close friend and colleague.

We hope readers find this encyclopedia a useful and valuable compendium and we welcome your feedback.

Boston, USA   John Baillieul
Minneapolis, USA   Tariq Samad
May 2015

Section Editors

Linear Systems Theory (Time-Domain)
Panos J. Antsaklis Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Aerospace Applications
Gary Balas Deceased. Formerly at Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Game Theory
Tamer Başar Coordinated Science Laboratory, University of Illinois, Urbana, IL, USA

Economic and Financial Systems
Alain Bensoussan Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX, USA

Geometric Optimal Control
Anthony Bloch Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Classical Optimal Control
Michael Cantoni Department of Electrical & Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

Discrete-Event Systems
Christos G. Cassandras Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Electric Energy Systems
Joe H. Chow Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Control of Networked Systems
Jorge Cortés Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Estimation and Filtering
Frederick E. Daum Raytheon Company, Woburn, MA, USA

Control of Process Systems
Sebastian Engell Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Automotive and Road Transportation
Luigi Glielmo Facoltà di Ingegneria dell'Università del Sannio in Benevento, Benevento, Italy

Stochastic Control
Lei Guo Academy of Mathematics and Systems Science, Chinese Academy of Sciences (CAS), Beijing, China

Nonlinear Control
Alberto Isidori Department of Computer and System Sciences "A. Ruberti", University of Rome "La Sapienza", Rome, Italy

Biosystems and Control
Mustafa Khammash Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

Distributed Parameter Systems
Miroslav Krstic Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Hybrid Systems
Francoise Lamnabhi-Lagarrigue Laboratoire des Signaux et Systèmes – CNRS, European Embedded Control Institute, Gif-sur-Yvette, France

Identification and Modeling
Lennart Ljung Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Model-Predictive Control
David Mayne Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Adaptive Control
Richard Hume Middleton School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Intelligent Control
Thomas Parisini Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Control of Marine Vessels
Kristin Y. Pettersen Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

Frequency-Domain Control
Michael Sebek Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 6, Czech Republic

Robotics
Bruno Siciliano Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Napoli, Italy

Other Applications of Advanced Control
Toshiharu Sugie Department of Systems Science, Graduate School of Informatics, Kyoto University, Uji, Kyoto, Japan

Complex Systems with Uncertainty
Roberto Tempo CNR-IEIIT, Politecnico di Torino, Torino, Italy

Control of Manufacturing Systems
Dawn Tilbury Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI, USA

Computer-Aided Control Systems Design
Andreas Varga Institute of System Dynamics and Control, German Aerospace Center, DLR Oberpfaffenhofen, Wessling, Germany

Information-Based Control
Wing-Shing Wong Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

Robust Control
Kemin Zhou Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA

Contributors

Teodoro Alamo Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Frank Allgöwer Institute for Systems Theory and Automatic Control, University of Stuttgart, Stuttgart, Germany

Tansu Alpcan Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, Australia

Eitan Altman INRIA, Sophia-Antipolis, France

Sean B. Andersson Mechanical Engineering and Division of Systems Engineering, Boston University, Boston, MA, USA

David Angeli Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria dell'Informazione, University of Florence, Italy

Finn Ankersen European Space Agency, Noordwijk, The Netherlands

Anuradha M. Annaswamy Active-adaptive Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Gianluca Antonelli University of Cassino and Southern Lazio, Cassino, Italy

Panos J. Antsaklis Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Pierre Apkarian DCSD, ONERA – The French Aerospace Lab, Toulouse, France

A. Astolfi Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

Karl Åström Department of Automatic Control, Lund University, Lund, Sweden

Thomas A. Badgwell ExxonMobil Research & Engineering, Annandale, NJ, USA

John Baillieul Mechanical Engineering, Electrical and Computer Engineering, Boston University, Boston, MA, USA

Gary Balas Deceased. Formerly at Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Yaakov Bar-Shalom University of Connecticut, Storrs, CT, USA

B. Ross Barmish University of Wisconsin, Madison, WI, USA

Tamer Başar Coordinated Science Laboratory, University of Illinois, Urbana, IL, USA

Georges Bastin Department of Mathematical Engineering, University Catholique de Louvain, Louvain-La-Neuve, Belgium

Karine Beauchard CNRS, CMLS, Ecole Polytechnique, Palaiseau, France

Nikolaos Bekiaris-Liberis Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Christine M. Belcastro NASA Langley Research Center, Hampton, VA, USA

Alberto Bemporad IMT Institute for Advanced Studies Lucca, Lucca, Italy

Peter Benner Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Pierre Bernhard INRIA-Sophia Antipolis Méditerranée, Sophia Antipolis, France

Tomasz R. Bielecki Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA

Mogens Blanke Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark (DTU), Lyngby, Denmark

Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, Trondheim, Norway

Anthony Bloch Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Bernard Bonnard Institute of Mathematics, University of Burgundy, Dijon, France

Dominique Bonvin Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

Ugo Boscain CNRS CMAP, École Polytechnique, Palaiseau, France

Team GECO INRIA Saclay, Palaiseau, France

James E. Braun Purdue University, West Lafayette, IN, USA

Roger Brockett Harvard University, Cambridge, MA, USA

Linda Bushnell Department of Electrical Engineering, University of Washington, Seattle, WA, USA

Fabrizio Caccavale School of Engineering, Università degli Studi della Basilicata, Potenza, Italy

Abel Cadenillas University of Alberta, Edmonton, AB, Canada

Peter E. Caines McGill University, Montreal, QC, Canada

Andrea Caiti DII – Department of Information Engineering & Centro "E. Piaggio", ISME – Interuniversity Research Centre on Integrated Systems for the Marine Environment, University of Pisa, Pisa, Italy

Octavia Camps Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA

Mark Cannon Department of Engineering Science, University of Oxford, Oxford, UK

Michael Cantoni Department of Electrical & Electronic Engineering, The University of Melbourne, Parkville, VIC, Australia

Xi-Ren Cao Department of Finance and Department of Automation, Shanghai Jiao Tong University, Shanghai, China

Institute of Advanced Study, Hong Kong University of Science and Technology, Hong Kong, China

Daniele Carnevale Dipartimento di Ing. Civile ed Ing. Informatica, Università di Roma "Tor Vergata", Roma, Italy

Giuseppe Casalino University of Genoa, Genoa, Italy

Francesco Casella Politecnico di Milano, Milan, Italy

Christos G. Cassandras Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

David A. Castañón Boston University, Boston, MA, USA

Eduardo Cerpa Departamento de Matemática, Universidad Técnica Federico Santa María, Valparaiso, Chile

François Chaumette Inria, Rennes, France

Ben M. Chen Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore

Jie Chen City University of Hong Kong, Hong Kong, China

Tongwen Chen Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada

Benoit Chevalier-Roignant Oliver Wyman, Munich, Germany

Hsiao-Dong Chiang School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA

Stefano Chiaverini Dipartimento di Ingegneria Elettrica e dell'Informazione "Maurizio Scarano", Università degli Studi di Cassino e del Lazio Meridionale, Cassino (FR), Italy

Luigi Chisci Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Firenze, Italy

Alessandro Chiuso Department of Information Engineering, University of Padova, Padova, Italy

Joe H. Chow Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Monique Chyba University of Hawaii-Manoa, Manoa, HI, USA

Martin Corless School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN, USA

Jean-Michel Coron Laboratoire Jacques-Louis Lions, University Pierre et Marie Curie, Paris, France

Jorge Cortés Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Fabrizio Dabbene CNR-IEIIT, Politecnico di Torino, Torino, Italy

Mark L. Darby CMiD Solutions, Houston, TX, USA

Frederick E. Daum Raytheon Company, Woburn, MA, USA

David Martin De Diego Instituto de Ciencias Matemáticas (CSIC-UAM-UC3M-UCM), Madrid, Spain

Alessandro De Luca Sapienza Università di Roma, Roma, Italy

Cesar de Prada Departamento de Ingeniería de Sistemas y Automática, University of Valladolid, Valladolid, Spain

Enrique Del Castillo Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA, USA

Luigi del Re Johannes Kepler Universität, Linz, Austria

Domitilla Del Vecchio Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Moritz Diehl Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany

ESAT-STADIUS/OPTEC, KU Leuven, Leuven-Heverlee, Belgium

Steven X. Ding University of Duisburg-Essen, Duisburg, Germany

Ian Dobson Iowa State University, Ames, IA, USA

Alejandro D. Domínguez-García University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, USA

Sebastian Dormido Departamento de Informatica y Automatica, UNED, Madrid, Spain

Tyrone Duncan Department of Mathematics, University of Kansas, Lawrence, KS, USA

Alexander Efremov Moscow Aviation Institute, Moscow, Russia

Magnus Egerstedt Georgia Institute of Technology, Atlanta, GA, USA

Naomi Ehrich Leonard Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA

Abbas Emami-Naeini Stanford University, Stanford, CA, USA

Sebastian Engell Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Dale Enns Honeywell International Inc., Minneapolis, MN, USA

Kaan Erkorkmaz Department of Mechanical & Mechatronics Engineering, University of Waterloo, Waterloo, ON, Canada

Fabio Fagnani Dipartimento di Scienze Matematiche 'G.L. Lagrange', Politecnico di Torino, Torino, Italy

Maurizio Falcone Dipartimento di Matematica, SAPIENZA – Università di Roma, Rome, Italy

Paolo Falcone Department of Signals and Systems, Mechatronics Group, Chalmers University of Technology, Göteborg, Sweden

Alfonso Farina Selex ES, Roma, Italy

Heike Faßbender Institut Computational Mathematics, Technische Universität Braunschweig, Braunschweig, Germany

Augusto Ferrante Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy

Thor I. Fossen Department of Engineering Cyberentics, Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Trondheim, Norway

Bruce A. Francis Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Emilio Frazzoli Massachusetts Institute of Technology, Cambridge, MA, USA

Georg Frey Saarland University, Saarbrücken, Germany

Minyue Fu School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW, Australia

Sergio Galeani Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma "Tor Vergata", Roma, Italy

Mario Garcia-Sanz Case Western Reserve University, Cleveland, OH, USA

Janos Gertler George Mason University, Fairfax, VA, USA

Alessandro Giua DIEE, University of Cagliari, Cagliari, Italy

LSIS, Aix-en-Provence, France

S. Torkel Glad Department of Electrical Engineering, Linköping University, Linköping, Sweden

Keith Glover Department of Engineering, University of Cambridge, Cambridge, UK

Ambarish Goswami Honda Research Institute, Mountain View, CA, USA

Pulkit Grover Carnegie Mellon University, Pittsburgh, PA, USA

Lars Grüne Mathematical Institute, University of Bayreuth, Bayreuth, Germany

Vijay Gupta Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Fredrik Gustafsson Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Christoforos N. Hadjicostis University of Cyprus, Nicosia, Cyprus

Tore Hägglund Lund University, Lund, Sweden

Christine Haissig Honeywell International Inc., Minneapolis, MN, USA

Bruce Hajek University of Illinois, Urbana, IL, USA

Alain Haurie ORDECSYS and University of Geneva, Switzerland

GERAD-HEC Montréal PQ, Canada

W.P.M.H. Heemels Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

Didier Henrion LAAS-CNRS, University of Toulouse, Toulouse, France

Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

João P. Hespanha Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA

Håkan Hjalmarsson School of Electrical Engineering, ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden

Ying Hu IRMAR, Université Rennes 1, Rennes Cedex, France

Luigi Iannelli Università degli Studi del Sannio, Benevento, Italy

Pablo A. Iglesias Electrical & Computer Engineering, The Johns Hopkins University, Baltimore, MD, USA

Petros A. Ioannou University of Southern California, Los Angeles, CA, USA

Hideaki Ishii Tokyo Institute of Technology, Yokohama, Japan

Alberto Isidori Department of Computer and System Sciences "A. Ruberti", University of Rome "La Sapienza", Rome, Italy

Tetsuya Iwasaki Department of Mechanical & Aerospace Engineering, University of California, Los Angeles, CA, USA

Ali Jadbabaie University of Pennsylvania, Philadelphia, PA, USA

Monique Jeanblanc Laboratoire Analyse et Probabilités, IBGBI, Université d'Evry Val d'Essonne, Evry Cedex, France

Karl H. Johansson ACCESS Linnaeus Center, Royal Institute of Technology, Stockholm, Sweden

Ramesh Johari Stanford University, Stanford, CA, USA

Mihailo R. Jovanović Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

Hans-Michael Kaltenbach ETH Zürich, Basel, Switzerland

Christoph Kawan Courant Institute of Mathematical Sciences, New York University, New York, USA

Matthias Kawski School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA

Hassan K. Khalil Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

Mustafa Khammash Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

Rudibert King Technische Universität Berlin, Berlin, Germany

Basil Kouvaritakis Department of Engineering Science, University of Oxford, Oxford, UK

A.J. Krener Department of Applied Mathematics, Naval Postgrauate School, Monterey, CA, USA

Miroslav Krstic Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Vladimír Kučera Faculty of Electrical Engineering, Czech Technical University of Prague, Prague, Czech Republic

Stéphane Lafortune Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Frank L. Lewis Arlington Research Institute, University of Texas, Fort Worth, TX, USA

Daniel Limon Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Kang-Zhi Liu Department of Electrical and Electronic Engineering, Chiba University, Chiba, Japan

Lennart Ljung Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Marco Lovera Politecnico di Milano, Milan, Italy

J. Lygeros Automatic Control Laboratory, Swiss Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland

Kevin M. Lynch Mechanical Engineering Department, Northwestern University, Evanston, IL, USA

Ronald Mahler Eagan, MN, USA

Lorenzo Marconi C.A.SY. Ű- DEI, University of Bologna, Bologna, Italy

Sonia Martínez Department of Mechanical and Aerospace Engineering, University of California, La Jolla, San Diego, CA, USA

Wolfgang Mauntz Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Volker Mehrmann Institut für Mathematik MA 4-5, Technische Universität Berlin, Berlin, Germany

Claudio Melchiorri Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Alma Mater Studiorum Università di Bologna, Bologna, Italy

Mehran Mesbahi University of Washington, Seattle, WA, USA

Alexandre R. Mesquita Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

Thomas Meurer Faculty of Engineering, Christian-Albrechts-University Kiel, Kiel, Germany

Wim Michiels KU Leuven, Leuven (Heverlee), Belgium

Richard Hume Middleton School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

S.O. Reza Moheimani School of Electrical Engineering & Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Julian Morris School of Chemical Engineering and Advanced Materials, Centre for Process Analytics and Control Technology, Newcastle University, Newcastle Upon Tyne, UK

James Moyne Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA

Curtis P. Mracek Raytheon Missile Systems, Waltham, MA, USA

Richard M. Murray Control and Dynamical Systems, Caltech, Pasadena, CA, USA

Hideo Nagai Osaka University, Osaka, Japan

Girish N. Nair Department of Electrical & Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia

Angelia Nedić Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL, USA

Arye Nehorai Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA

Dragan Nesic Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia

Brett Ninness School of Electrical and Computer Engineering, University of Newcastle, Newcastle, Australia

Hidekazu Nishimura Graduate School of System Design and Management, Keio University, Yokohama, Japan

Dominikus Noll Institut de Mathématiques, Université de Toulouse, Toulouse, France

Lorenzo Ntogramatzidis Department of Mathematics and Statistics, Curtin University, Perth, WA, Australia

Giuseppe Oriolo Sapienza Università di Roma, Roma, Italy

Richard W. Osborne University of Connecticut, Storrs, CT, USA

Martin Otter Institute of System Dynamics and Control, German Aerospace Center (DLR), Wessling, Germany

David H. Owens University of Sheffield, Sheffield, UK

Hitay Özbay Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey

Asuman Ozdaglar Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA

Andrew Packard Mechanical Engineering Department, University of California, Berkeley, CA, USA

Fernando Paganini Universidad ORT Uruguay, Montevideo, Uruguay

Gabriele Pannocchia University of Pisa, Pisa, Italy

Lucy Y. Pao University of Colorado, Boulder, CO, USA

George J. Pappas Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

Frank C. Park Robotics Laboratory, Seoul National University, Seoul, Korea

Bozenna Pasik-Duncan Department of Mathematics, University of Kansas, Lawrence, KS, USA

Ron J. Patton School of Engineering, University of Hull, Hull, UK

Marco Pavone Stanford University, Stanford, CA, USA

Shige Peng Shandong University, Jinan, Shandong Province, China

Tristan Perez Electrical Engineering & Computer Science, Queensland University of Technology, Brisbane, QLD, Australia

Ian R. Petersen School of Engineering and Information Technology, University of New South Wales, the Australian Defence Force Academy, Canberra, Australia

Kristin Y. Pettersen Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

Benedetto Piccoli Mathematical Sciences and Center for Computational and Integrative Biology, Rutgers University, Camden, NJ, USA

Rik Pintelon Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Boris Polyak Institute of Control Science, Moscow, Russia

Romain Postoyan Université de Lorraine, CRAN, France

CNRS, CRAN, France

J. David Powell Stanford University, Stanford, CA, USA

Laurent Praly MINES ParisTech, PSL Research University, CAS, Fontainebleau, France

Domenico Prattichizzo University of Siena, Siena, Italy

James A. Primbs University of Texas at Dallas, Richardson, TX, USA

S. Joe Qin University of Southern California, Los Angeles, CA, USA

Li Qiu Hong Kong University of Science and Technology, Hong Kong SAR, China

Rajesh Rajamani Department of Mechanical Engineering, University of Minnesota, Twin Cities, Minneapolis, MN, USA

Saša Raković Oxford University, Oxford, UK

James B. Rawlings University of Wisconsin, Madison, WI, USA

Jean-Pierre Raymond Institut de Mathématiques, Université Paul Sabatier Toulouse III & CNRS, Toulouse Cedex, France

Wei Ren Department of Electrical Engineering, University of California, Riverside, CA, USA

Spyros Reveliotis School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Giorgio Rizzoni Department of Mechanical and Aerospace Engineering, Center for Automotive Research, The Ohio State University, Columbus, OH, USA

L.C.G. Rogers University of Cambridge, Cambridge, UK

Pierre Rouchon Centre Automatique et Systèmes, Mines ParisTech, Paris Cedex 06, France

Ricardo G. Sanfelice Department of Computer Engineering, University of California at Santa Cruz, Santa Cruz, CA, USA

Heinz Schättler Washington University, St. Louis, MO, USA

Thomas B. Schön Department of Information Technology, Uppsala University, Uppsala, Sweden

Johan Schoukens Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Michael Sebek Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague 6, Czech Republic

Peter Seiler Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Suresh P. Sethi Jindal School of Management, The University of Texas at Dallas, Richardson, TX, USA

Sirish L. Shah Department of Chemical and Materials Engineering, University of Alberta Edmonton, Edmonton, AB, Canada

Jeff S. Shamma School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Pavel Shcherbakov Institute of Control Science, Moscow, Russia

Jianjun Shi Georgia Institute of Technology, Atlanta, GA, USA

Manuel Silva Instituto de Investigation en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain

Vasile Sima Advanced Research, National Institute for Research & Development in Informatics, Bucharest, Romania

Robert E. Skelton University of California, San Diego, CA, USA

Sigurd Skogestad Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Eduardo D. Sontag   Rutgers University, New Brunswick, NJ, USA

Asgeir J. Sørensen Department of Marine Technology, Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, NTNU, Trondheim, Norway

Mark W. Spong Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas, Richardson, TX, USA

R. Srikant Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, Champaign, IL, USA

Jörg Stelling   ETH Zürich, Basel, Switzerland

Jing Sun   University of Michigan, Ann Arbor, MI, USA

Krzysztof Szajowski Faculty of Fundamental Problems of Technology, Institute of Mathematics and Computer Science, Wroclaw University of Technology, Wroclaw, Poland

Mario Sznaier Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA

Paulo Tabuada Department of Electrical Engineering, University of California, Los Angeles, CA, USA

Satoshi Tadokoro   Tohoku University, Sendai, Japan

Gongguo Tang   Department of Electrical Engineering & Computer Science, Colorado School of Mines, Golden, CO, USA

Shanjian Tang   Fudan University, Shanghai, China

Andrew R. Teel Electrical and Computer Engineering Department, University of California, Santa Barbara, CA, USA

Roberto Tempo   CNR-IEIIT, Politecnico di Torino, Torino, Italy

Onur Toker   Fatih University, Istanbul, Turkey

H.L. Trentelman Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Jorge Otávio Trierweiler   Group of Intensification, Modelling, Simulation, Control and Optimization of Processes (GIMSCOP), Department of Chemical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil

Lenos Trigeorgis   University of Cyprus, Nicosia, Cyprus

Eric Tseng   Ford Motor Company, Dearborn, MI, USA

Kyriakos G. Vamvoudakis Center for Control, Dynamical Systems and Computation (CCDC), University of California, Santa Barbara, CA, USA

Paul Van Dooren ICTEAM: Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium

O. Arda Vanli Department of Industrial and Manufacturing Engineering, High Performance Materials Institute Florida A&M University and Florida State University, Tallahassee, FL, USA

Andreas Varga Institute of System Dynamics and Control, German Aerospace Center, DLR Oberpfaffenhofen, Wessling, Germany

Michel Verhaegen Delft Center for Systems and Control, Delft University, Delft, The Netherlands

Mathukumalli Vidyasagar University of Texas at Dallas, Richardson, TX, USA

Luigi Villani Dipartimento di Ingeneria Elettrica e Tecnologie dell' Informazione, Università degli Studi di Napoli Federico II, Napoli, Italy

Richard B. Vinter Imperial College, London, UK

Antonio Visioli Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia, Italy

Vijay Vittal Arizona State University, Tempe, AZ, USA

Costas Vournas School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Greece

Steffen Waldherr Institute for Automation Engineering, Otto-von-Guericke-Universität Magdgeburg, Magdeburg, Germany

Fred Wang University of Tennessee, Knoxville, TN, USA

Yorai Wardi School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

John M. Wassick The Dow Chemical Company, Midland, MI, USA

Wing-Shing Wong Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

W.M. Wonham Department of Electrical & Computer Engineering, University of Toronto, Toronto, ON, Canada

Yutaka Yamamoto Department of Applied Analysis and Complex Dynamical Systems, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Hong Ye The Mathworks, Inc., Natick, MA, USA

George Yin Department of Mathematics, Wayne State University, Detroit, MI, USA

Serdar Yüksel Department of Mathematics and Statistics, Queen's University, Kingston, ON, Canada

Michael M. Zavlanos Department of Mechanical Engineering and
Materials Science, Duke University, Durham, NC, USA

Qing Zhang Department of Mathematics, The University of Georgia,
Athens, GA, USA

Qinghua Zhang Inria, Campus de Beaulieu, Rennes Cedex, France

A

Active Power Control of Wind Power Plants for Grid Integration

Lucy Y. Pao
University of Colorado, Boulder, CO, USA

Abstract

Increasing penetrations of intermittent renewable energy sources, such as wind, on the utility grid have led to concerns over the reliability of the grid. One approach for improving grid reliability with increasing wind penetrations is to actively control the real power output of wind turbines and wind power plants. Providing a full range of responses requires derating wind power plants so that there is headroom to both increase and decrease power to provide grid balancing services and stabilizing responses. Initial results indicate that wind turbines may be able to provide primary frequency control and frequency regulation services more rapidly than conventional power plants.

Keywords

Active power control; Automatic generation control; Frequency regulation; Grid balancing; Grid integration; Primary frequency control; Wind energy

Balancing Electrical Generation and Load on the Grid

Wind penetration levels across the world have increased dramatically, with installed capacity growing at a mean annual rate of 25 % over the last decade (Gsanger and Pitteloud 2013). Some nations in Western Europe, particularly Denmark, Portugal, Spain, and Germany, have seen wind provide more than 16 % of their annual electrical energy needs (Wiser and Bolinger 2013). To maintain grid frequency at its nominal value, the electrical generation must equal the electrical load on the grid. This balancing has historically been left up to conventional utilities with synchronous generators, which can vary their active power output by simply varying their fuel input. Grid frequency control is performed across a number of regimes and time scales, with both manual and automatic control commands. Further details can be found in Rebours et al. (2007) and Ela et al. (2011).

Wind turbines and wind power plants are now being recognized as having the potential to meet demanding grid stabilizing requirements set by transmission system operators (Aho et al. 2013a,b; Buckspan et al. 2012; Ela et al. 2011; Miller et al. 2011). Recent grid code requirements have spurred the development of wind turbine active power control (APC) systems, which allow wind turbines to participate in grid frequency regulation and provide stabilizing responses to

sudden changes in grid frequency. The ability of wind turbines to provide APC services also allows them to follow forecast-based power production schedules.

For a wind turbine to fully participate in grid frequency control, it must be derated (to P_{derated}) with respect to the maximum power (P_{max}) that can be generated given the available wind, allowing for both increases and decreases in power, if necessary. Wind turbines can derate their power output by pitching their blades to shed aerodynamic power or reducing their generator torque in order to operate at higher-than-optimal rotor speeds. Wind turbines can then respond at different time scales to provide more or less power through pitch control (which can provide a power response within seconds) and generator torque control (which can provide a power response within milliseconds).

Wind Turbine Inertial and Primary Frequency Control

Inertial and primary frequency control is generally considered to be the first 5–10 s after a frequency event occurs. In this regime, the governors of capable utilities actuate, allowing for a temporary increase or decrease in the utilities' power outputs. The primary frequency control (PFC) response provided by conventional synchronous generators can be characterized by a droop curve, which relates fluctuations in grid frequency to a change in power from the utility. For example, a 3 % droop curve means that a 3 % change in grid frequency yields a 100 % change in commanded power.

Although modern wind turbines do not inherently provide inertial or primary frequency control responses because their power electronics impart a buffer between their generators and the grid, such responses can be produced through careful design of the wind turbine control systems. While the physical properties of a conventional synchronous generator yield a static droop characteristic, a wind turbine can be controlled to provide a primary frequency response via either a static or time-varying droop curve.
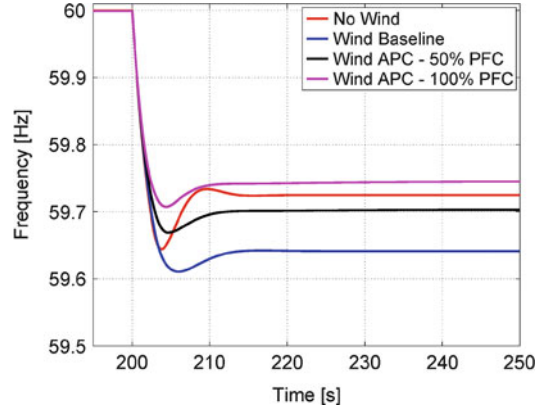
Active Power Control of Wind Power Plants for Grid Integration, Fig. 1 Simulation results showing the capability of wind power plants to provide APC services on a small-scale grid model. The total grid size is 3 GW, and a frequency event is induced due to the sudden active power imbalance when 5 % of generation is taken offline at time = 200 s. Each wind power plant is derated to 90 % of its rated capacity. The system response with all conventional generation (no wind) is compared to the cases when there are wind power plants on the grid at 10 % penetration (i) with a baseline control system (wind baseline) where wind does not provide APC services and (ii) with an APC system (wind APC) that uses a 3 % droop curve where either 50 % or 100 % of the wind power plants provide PFC

A time-varying droop curve can be designed to be more aggressive when the magnitude of the rate of change of frequency of the grid is larger.

Figure 1 shows a simulation of a grid response under different scenarios when 5 % of the generating capacity suddenly goes offline. When the wind power plant (10 % of the generation on the grid) is operating with its normal baseline control system that does not provide APC services, the frequency response is worse than the no-wind scenario, due to the reduced amount of conventional generation in the wind-baseline scenario that can provide power control services. However, compared to both the no-wind and wind-baseline cases, using PFC with a droop curve results in the frequency decline being arrested at a minimum (nadir) frequency f_{nadir} that is closer to the nominal $f_{\text{nom}} = 60$ Hz frequency level; further, the steady-state frequency f_{ss} after the PFC response is also closer to f_{nom}. It is important to prevent the difference $f_{\text{nom}} - f_{\text{nadir}}$

from exceeding a threshold that can lead to underfrequency load shedding (UFLS) or rolling blackouts. The particular threshold varies across utility grids, but the largest such threshold in North America is 1.5 Hz.

Stability issues arising from the altered control algorithms must be analyzed (Buckspan et al. 2013). The trade-offs between aggressive primary frequency control and resulting structural loads also need to be evaluated carefully. Initial research shows that potential grid support can be achieved while not causing any increases in structural loading and hence fatigue damage and operations and maintenance costs (Buckspan et al. 2012).

Wind Turbine Automatic Generation Control

Secondary frequency control, also known as automatic generation control (AGC), occurs on a slower time scale than PFC. AGC commands can be generated from highly damped proportional integral (PI) controllers or logic controllers to regulate grid frequency and are used to control the power output of participating power plants. In many geographical regions, frequency regulation services are compensated through a competitive market, where power plants that provide faster and more accurate AGC command tracking are paid more.

An active power control system that combines both primary and secondary/AGC frequency control capabilities has recently been detailed in Aho et al. (2013a). Figure 2 presents initial experimental field test results of this active power controller, in response to prerecorded frequency events, showing how responsive wind turbines can be to both manual derating commands as well as rapidly changing automatic primary frequency control commands generated via a droop curve. Overall, results indicate that wind turbines can respond more rapidly than conventional power plants. However, increasing the power control and regulation performance of a wind turbine should be carefully considered due to a number of complicating factors, including coupling with

existing control loops, a desire to limit actuator usage and structural loading, and wind variability.

Active Power Control of Wind Power Plants

A wind power plant, often referred to as a wind farm, consists of many wind turbines. In wind power plants, wake effects can reduce generation in downstream turbines to less than 60 % of the lead turbine (Barthelmie et al. 2009; Porté-Agel et al. 2013). There are many emerging areas of active research, including the modeling of wakes and wake effects and how these models can then be used to coordinate the control of individual turbines so that the overall wind power plant can reliably track the desired power reference command. A wind farm controller can be interconnected with the utility grid, transmission system operator (TSO), and individual turbines as shown in Fig. 3. By properly accounting for the wakes, wind farm controllers can allocate appropriate power reference commands to the individual wind turbines. Individual turbine generator torque and blade pitch controllers, as discussed earlier, can be designed so that each turbine follows the power reference command issued by the wind farm controller. Methods for intelligent, distributed control of entire wind farms to rapidly respond to grid frequency disturbances could significantly reduce frequency deviations and improve recovery speed to such disturbances.

Combining Techniques with Other Approaches for Balancing the Grid

Ultimately, active power control of wind turbines and wind power plants should be combined with both demand-side management and storage to provide a more comprehensive solution that enables balancing electrical generation and electrical load with large penetrations of wind energy on the grid. Demand-side management (Callaway and Hiskens 2011; Kowli and Meyn 2011; Palensky and Dietrich 2011)
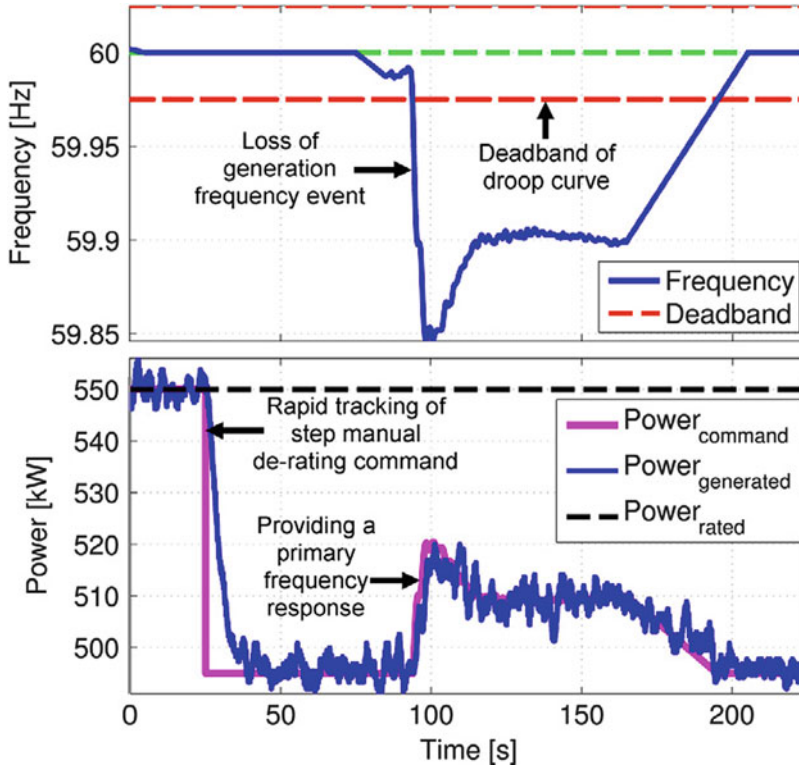
Active Power Control of Wind Power Plants for Grid Integration, Fig. 2 The frequency data input and power that is commanded and generated during a field test with a 550 kW research wind turbine at the US National Renewable Energy Laboratory (NREL). The frequency data was recorded on the Electric Reliability Council of Texas (ERCOT) interconnection (data courtesy of Vahan Gevorgian, NREL). The upper plot shows the grid frequency, which is passed through a 5 % droop curve with a deadband to generate a power command. The high-frequency fluctuations in the generated power would be smoothed when aggregating the power output of an entire wind power plant
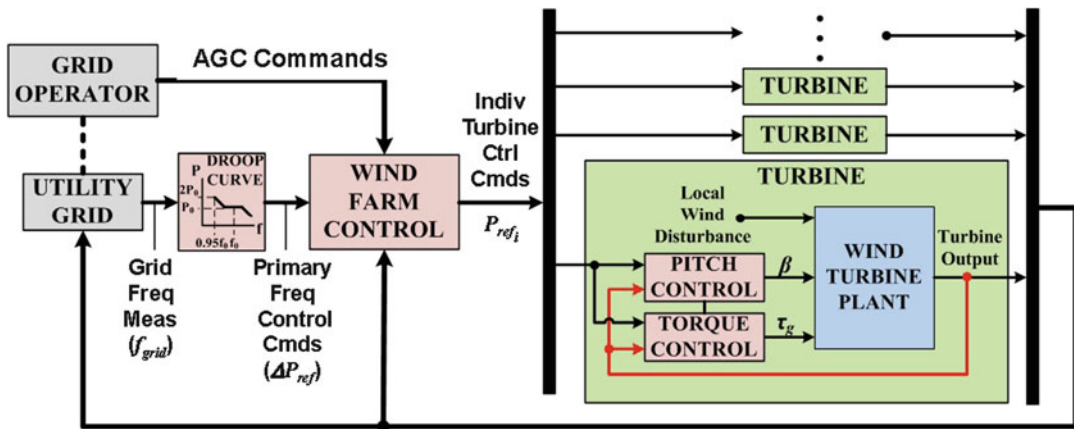
Active Power Control of Wind Power Plants for Grid Integration, Fig. 3 Schematic showing the communication and coupling between the wind farm control system, individual wind turbines, utility grid, and the grid operator. The wind farm controller uses measurements of the utility grid frequency and automatic generation control power command signals from the grid operator to determine a power reference for each turbine in the wind farm

aims to alter the demand in order to mitigate peak electrical loads and hence to maintain sufficient control authority among generating units. As more effective and economical energy storage solutions (Pickard and Abbott 2012) at the power plant scale are developed, wind (and solar) energy can then be stored when wind (and solar) energy availability is not well matched with electrical demand. Advances in wind forecasting (Giebel et al. 2011) will also improve wind power forecasts to facilitate more accurate scheduling of larger amounts of wind power on the grid.

Cross-References

▶ Control of Fluids and Fluid-Structure Interactions
▶ Control Structure Selection
▶ Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms
▶ Electric Energy Transfer and Control via Power Electronics
▶ Networked Control Systems: Architecture and Stability Issues
▶ Power System Voltage Stability
▶ Small Signal Stability in Electric Power Systems

Recommended Reading

A recent comprehensive report on active power control that covers topics ranging from control design to power system engineering to economics can be found in Ela et al. (2014) and the references therein.

Bibliography

Aho J, Pao L, Buckspan A, Fleming P (2013a) An active power control system for wind turbines capable of primary and secondary frequency control for supporting grid reliability. In: Proceedings of the AIAA aerospace sciences meeting, Grapevine, Jan 2013

Aho J, Buckspan A, Dunne F, Pao LY (2013b) Controlling wind energy for utility grid reliability. ASME Dyn Syst Control Mag 1(3):4–12

Barthelmie RJ, Hansen K, Frandsen ST, Rathmann O, Schepers JG, Schlez W, Phillips J, Rados K, Zervos A, Politis ES, Chaviaropoulos PK (2009) Modelling and measuring flow and wind turbine wakes in large wind farms offshore. Wind Energy 12:431–444

Buckspan A, Aho J, Fleming P, Jeong Y, Pao L (2012) Combining droop curve concepts with control systems for wind turbine active power control. In: Proceedings of the IEEE symposium power electronics and machines in wind applications, Denver, July 2012

Buckspan A, Pao L, Aho J, Fleming P (2013) Stability analysis of a wind turbine active power control system. In: Proceedings of the American control conference, Washington, DC, June 2013, pp 1420–1425

Callaway DS, Hiskens IA (2011) Achieving controllability of electric loads. Proc IEEE 99(1): 184–199

Ela E, Milligan M, Kirby B (2011) Operating reserves and variable generation. Technical report, National Renewable Energy Laboratory, NREL/TP-5500-51928

Ela E, Gevorgian V, Fleming P, Zhang YC, Singh M, Muljadi E, Scholbrock A, Aho J, Buckspan A, Pao L, Singhvi V, Tuohy A, Pourbeik P, Brooks D, Bhatt N (2014) Active power controls from wind power: bridging the gaps. Technical report, National Renewable Energy Laboratory, NREL/TP-5D00-60574, Jan 2014

Giebel G, Brownsword R, Kariniotakis G, Denhard M, Draxl C (2011) The state-of-the-art in short-term prediction of wind power: a literature overview. Technical report, ANEMOS.plus/SafeWind, Jan 2011

Gsanger S, Pitteloud J-D (2013) World wind energy report 2012. The World Wind Energy Association, May 2013

Kowli AS, Meyn SP (2011) Supporting wind generation deployment with demand response. In: Proceedings of the IEEE power and energy society general meeting, Detroit, July 2011

Miller N, Clark K, Shao M (2011) Frequency responsive wind plant controls: impacts on grid performance. In: Proceedings of the IEEE power and energy society general meeting, Detroit, July 2011

Palensky P, Dietrich D (2011) Demand-side management: demand response, intelligent energy systems, and smart loads. IEEE Trans Ind Inform 7(3): 381–388

Pickard WF, Abbott D (eds) (2012) The intermittency challenge: massive energy storage in a sustainable future. Proc IEEE 100(2):317–321. Special issue

Porté-Agel F, Wu Y-T, Chen C-H (2013) A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm. Energies 6:5297–5313

Rebours Y, Kirschen D, Trotignon M, Rossignol S (2007) A survey of frequency and voltage control ancillary services-part I: technical features. IEEE Trans Power Syst 22(1):350–357

Wiser R, Bolinger M (2013) 2012 Wind technologies market report. Lawrence Berkeley National Laboratory Report, Aug 2013

Adaptive Control of Linear Time-Invariant Systems

Petros A. Ioannou
University of Southern California, Los Angeles, CA, USA

Abstract

Adaptive control of linear time-invariant (LTI) systems deals with the control of LTI systems whose parameters are constant but otherwise completely unknown. In some cases, large norm bounds as to where the unknown parameters are located in the parameter space are also assumed to be known. In general, adaptive control deals with LTI plants which cannot be controlled with fixed gain controllers, i.e., nonadaptive control methods, and their parameters even though assumed constant for design and analysis purposes may change over time in an unpredictable manner. Most of the adaptive control approaches for LTI systems use the so-called certainty equivalence principle where a control law motivated from the known parameter case is combined with an adaptive law for estimating on line the unknown parameters. The control law could be associated with different control objectives and the adaptive law with different parameter estimation techniques. These combinations give rise to a wide class of adaptive control schemes. The two popular control objectives that led to a wide range of adaptive control schemes include model reference adaptive control (MRAC) and adaptive pole placement control (APPC). In MRAC, the control objective is for the plant output to track the output of a reference model, designed to represent the desired properties of the plant, for any reference input signal. APPC is more general and is based on control laws whose objective is to set the poles of the closed loop at desired locations chosen based on performance requirements. Another class of adaptive controllers for LTI systems that involves ideas from MRAC and APPC
is based on multiple models, search methods, and switching logic. In this class of schemes, the unknown parameter space is partitioned to smaller subsets. For each subset, a parameter estimator or a stabilizing controller is designed or a combination of the two. The problem then is to identify which subset in the parameter space the unknown plant model belongs to and/or which controller is a stabilizing one and meets the control objective. A switching logic is designed based on different considerations to identify the most appropriate plant model or controller from the list of candidate plant models and/or controllers. In this entry, we briefly describe the above approaches to adaptive control for LTI systems.

Keywords

Adaptive pole placement control; Direct MRAC; Indirect MRAC; LTI systems; Model reference adaptive control; Robust adaptive control

Model Reference Adaptive Control

In model reference control (MRC), the desired plant behavior is described by a reference model which is simply an LTI system with a transfer function $W_m(s)$ and is driven by a reference input. The controller transfer function $C(s, \theta^*)$, where θ^* is a vector with the coefficients of $C(s)$, is then developed so that the closed-loop plant has a transfer function equal to $W_m(s)$. This transfer function matching guarantees that the plant will match the reference model response for any reference input signal. In this case the plant transfer function $G_p(s, \theta_p^*)$, where θ_p^* is a vector with all the coefficients of $G_p(s)$, together with the controller transfer function $C(s, \theta^*)$ should lead to a closed-loop transfer function from the reference input r to the plant output y_p that is equal to $W_m(s)$, i.e.,

$$\frac{y_p(s)}{r(s)} = W_m(s) = \frac{y_m(s)}{r(s)}, \qquad (1)$$

where y_m is the output of the reference model. For this transfer matching to be possible, $G_p(s)$ and $W_m(s)$ have to satisfy certain assumptions. These assumptions enable the calculation of the controller parameter vector θ^* as

$$\theta^* = F(\theta_p^*), \qquad (2)$$

where F is a function of the plant parameters θ_p^*, to satisfy the matching equation (1). The function in (2) has a special form in the case of MRC that allows the design of both direct and indirect MRAC. For more general classes of controller structures, this is not possible in general as the function F is nonlinear. This transfer function matching guarantees that the tracking error $e_1 = y_p - y_m$ converges to zero for any given reference input signal r. If the plant parameter vector θ_p^* is known, then the controller parameters θ^* can be calculated using (2), and the controller $C(s, \theta^*)$ can be implemented. We are considering the case where θ_p^* is unknown. In this case, the use of the certainty equivalence (CE) approach, (Astrom and Wittenmark 1995; Egardt 1979; Ioannou and Fidan 2006; Ioannou and Kokotovic 1983; Ioannou and Sun 1996; Landau 1979; Landau et al. 1998; Morse 1996; Narendra and Annaswamy 1989; Narendra and Balakrishnan 1997; Sastry and Bodson 1989; Stefanovic and Safonov 2011; Tao 2003) where the unknown parameters are replaced with their estimates, leads to the adaptive control scheme referred to as *indirect MRAC*, shown in Fig. 1a.

The unknown plant parameter vector θ_p^* is estimated at each time t denoted by $\theta_p(t)$, using an online parameter estimator referred to as adaptive law. The plant parameter estimate $\theta_p(t)$ at each

time t is then used to calculate the controller parameter vector $\theta(t) = F(\theta_p(t))$ used in the controller $C(s, \theta)$. This class of MRAC is called indirect MRAC, because the controller parameters are not updated directly, but calculated at each time t using the estimated plant parameters. Another way of designing MRAC schemes is to parameterize the plant transfer function in terms of the desired controller parameter vector θ^*. This is possible in the MRC case, because the structure of the MRC law is such that we can use (2) to write

$$\theta_p^* = F^{-1}(\theta^*), \qquad (3)$$

where F^{-1} is the inverse of the mapping $F(\cdot)$, and then express $G_p(s, \theta_p^*) = G_p(s, F^{-1}(\theta^*)) = \bar{G}_p(s, \theta^*)$. The adaptive law for estimating θ^* online can now be developed by using $y_p = \bar{G}_p(s, \theta^*)u_p$ to obtain a parametric model that is appropriate for estimating the controller vector θ^* as the unknown parameter vector. The MRAC can then be developed using the CE approach as shown in Fig. 1b. In this case, the controller parameter $\theta(t)$ is updated directly without any intermediate calculations, and for this reason, the scheme is called *direct MRAC*.

The division of MRAC to indirect and direct is, in general, unique to MRC structures, and it is possible due to the fact that the inverse maps in (2) and (3) exist which is a direct consequence of the control objective and the assumptions the plant and reference model are required to satisfy for the control law to exist. These assumptions are summarized below:

Plant Assumptions: $G_p(s)$ is minimum phase, i.e., has stable zeros, its relative degree, $n^* =$
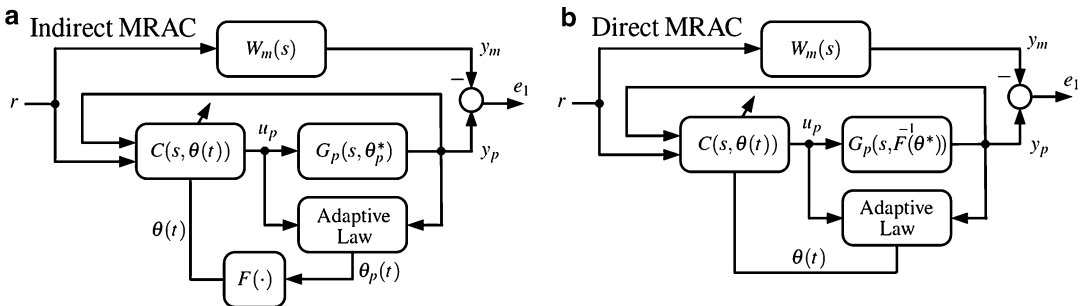
Adaptive Control of Linear Time-Invariant Systems, Fig. 1 Structure of (**a**) indirect MRAC, (**b**) direct MRAC

number of poles−number of zeros, is known and an upper bound n on its order is also known. In addition, the sign of its high-frequency gain is known even though it can be relaxed with additional complexity.

Reference Model Assumptions: $W_m(s)$ has stable poles and zeros, its relative degree is equal to n^* that of the plant, and its order is equal or less to the one assumed for the plant, i.e., of n.

The above assumptions are also used to meet the control objective in the case of known parameters, and therefore the minimum phase and relative degree assumptions are characteristics of the control objective and do not arise because of adaptive control considerations. The relative degree matching is used to avoid the need to differentiate signals in the control law. The minimum phase assumption comes from the fact that the only way for the control law to force the closed-loop plant transfer function to be equal to that of the reference model is to cancel the zeros of the plant using feedback and replace them with those of the reference model using a feedforward term. Such zero pole cancelations are possible if the zeros are stable, i.e., the plant is minimum phase; otherwise stability cannot be guaranteed for nonzero initial conditions and/or inexact cancelations.

The design of MRAC in Fig. 1 has additional variations depending on how the adaptive law is designed. If the reference model is chosen to be strictly positive real (SPR) which limits its transfer function and that of the plant to have relative degree 1, the derivation of adaptive law and stability analysis is fairly straightforward, and for this reason, this class of MRAC schemes attracted a lot of interest. As the relative degree changes to 2, the design becomes more complex as in order to use the SPR property, the CE control law has to be modified by adding an extra nonlinear term. The stability analysis remains to be simple as a single Lyapunov function can be used to establish stability. As the relative degree increases further, the design complexity increases by requiring the addition of more nonlinear terms in the CE control law (Ioannou and Fidan 2006; Ioannou and Sun 1996). The simplicity of using a single Lyapunov function analysis for stability

remains however. This approach covers both direct and indirect MRAC and lead to adaptive laws which contain no normalization signals (Ioannou and Fidan 2006; Ioannou and Sun 1996). A more straightforward design approach is based on the CE principle which separates the control design from the parameter estimation part and leads to a much wider class of MRAC which can be direct or indirect. In this case, the adaptive laws need to be normalized for stability, and the analysis is far more complicated than the approach based on SPR with no normalization. An example of such a direct MRAC scheme for the case of known sign of the high-frequency gain which is assumed to be positive for both plant and reference model is listed below:

Control law:

$$u_p = \theta_1^T(t) \frac{\alpha(s)}{\Lambda(s)} u_p + \theta_2^T \frac{\alpha(s)}{\Lambda(s)} y_p + \theta_3(t) y_p$$

$$+ c_0(t) r = \theta^T(t)\omega, \qquad (4)$$

where $\alpha \triangleq \alpha_{n-2}(s) = [s^{n-2}, s^{n-3}, \ldots, s, 1]^T$ for $n \geq 2$, and $\alpha(s) \triangleq 0$ for $n = 1$, and $\Lambda(s)$ is a monic polynomial with stable roots and degree $n - 1$ having numerator of $W_m(s)$ as a factor.

Adaptive law:

$$\dot{\theta} = \Gamma \varepsilon \phi, \qquad (5)$$

where Γ is a positive definite matrix referred to as the adaptive gain and $\dot{\rho} = \gamma \varepsilon \xi$, $\varepsilon = \frac{e_1 - \rho \xi}{m_s^2}$, $m_s^2 = 1 + \phi^T \phi + u_f^2$, $\xi = \theta^T \phi + u_f$, $\phi = -W_m(s)\omega$, and $u_f = W_m(s) u_p$.

The stability properties of the above direct MRAC scheme which are typical for all classes of MRAC are the following (Ioannou and Fidan 2006; Ioannou and Sun 1996): *(i) All signals in the closed-loop plant are bounded, and the tracking error e_1 converges to zero asymptotically and (ii) if the plant transfer function contains no zero pole cancelations and r is sufficiently rich of order $2n$, i.e., it contains at least n distinct frequencies, then the parameter error $|\tilde{\theta}| = |\theta - \theta^*|$ and the tracking error e_1 converge to zero exponentially fast.*

Adaptive Pole Placement Control

Let us consider the SISO LTI plant:

$$y_p = G_p(s)u_p, \quad G_p(s) = \frac{Z_p(s)}{R_p(s)}, \qquad (6)$$

where $G_p(s)$ is proper and $R_p(s)$ is a monic polynomial. The control objective is to choose the plant input u_p so that the closed-loop poles are assigned to those of a given monic Hurwitz polynomial $A^*(s)$, and y_p is required to follow a certain class of reference signals y_m assumed to satisfy $Q_m(s)y_m = 0$ where $Q_m(s)$ is known as the internal model of y_m and is designed to have all roots in $Re\{s\} \leq 0$ with no repeated roots on the $j\omega$-axis. The polynomial $A^*(s)$, referred to as the desired closed-loop characteristic polynomial, is chosen based on the closed-loop performance requirements. To meet the control objective, we make the following assumptions about the plant:

P1. $G_p(s)$ is strictly proper with known degree, and $R_p(s)$ is a monic polynomial whose degree n is known and $Q_m(s)Z_p(s)$ and $R_p(s)$ are coprime.

Assumption P1 allows Z_p and R_p to be non-Hurwitz in contrast to the MRAC case where Z_p is required to be Hurwitz.

The design of the APPC scheme is based on the CE principle. The plant parameters are estimated at each time t and used to calculate the controller parameters that meet the control objective for the estimated plant as follows: Using (6) the plant equation can be expressed in a form convenient for parameter estimation via the model (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996):

$$z = \theta_p^* \phi,$$

where $z = \frac{s^n}{\Lambda_p(s)}y_p$, $\theta_p^* = [\theta_b^{*T}, \theta_a^{*T}]^T$, $\phi = [\frac{\alpha_{n-1}^T(s)}{\Lambda_p(s)}u_p, -\frac{\alpha_{n-1}^T(s)}{\Lambda_p(s)}y_p]^T$, $\alpha_{n-1} = [s^{n-1}, \ldots, s, 1]^T$, $\theta_a^* = [a_{n-1}, \ldots, a_0]^T$, $\theta_b^* = [b_{n-1}, \ldots, b_0]^T$, and $\Lambda_p(s)$ is a Hurwitz monic design polynomial. As an example of a parameter estimation algorithm, we consider the gradient algorithm

$$\dot{\theta}_p = \Gamma \varepsilon \phi, \quad \varepsilon = \frac{z - \theta_p^T \phi}{m_s^2}, \quad m_s^2 = 1 + \phi^T \phi, \qquad (7)$$

where $\Gamma = \Gamma^T > 0$ is the adaptive gain and $\theta_p = [\hat{b}_{n-1}, \ldots, \hat{b}_0, \hat{a}_{n-1}, \ldots, \hat{a}_0]^T$ are the estimated plant parameters which can be used to form the estimated plant polynomials $\hat{R}_p(s, t) = s^n + \hat{a}_{n-1}(t)s^{n-1} + \ldots + \hat{a}_1(t)s + \hat{a}_0(t)$ and $\hat{Z}_p(s, t) = \hat{b}_{n-1}(t)s^{n-1} + \ldots + \hat{b}_1(t)s + \hat{b}_0(t)$ of $R_p(s)$ and $Z_p(s)$, respectively, at each time t. The adaptive control law is given as

$$u_p = \left(\Lambda(s) - \hat{L}(s, t)Q_m(s) \right) \frac{1}{\Lambda(s)}u_p$$
$$- \hat{P}(s, t)\frac{1}{\Lambda(s)}(y_p - y_m), \qquad (8)$$

where $\hat{L}(s, t)$ and $\hat{P}(s, t)$ are obtained by solving the polynomial equation $\hat{L}(s, t) \cdot Q_m(s) \cdot \hat{R}_p(s, t) + \hat{P}(s, t) \cdot \hat{Z}_p(s, t) = A^*(s)$ at each time t. The operation $X(s, t) \cdot Y(s, t)$ denotes a multiplication of polynomials where s is simply treated as a variable. The existence and uniqueness of $\hat{L}(s, t)$ and $\hat{P}(s, t)$ is guaranteed provided $\hat{R}_p(s, t) \cdot Q_m(s)$ and $\hat{Z}_p(s, t)$ are coprime at each frozen time t. The adaptive laws that generate the coefficients of $\hat{R}_p(s, t)$ and $\hat{Z}_p(s, t)$ cannot guarantee this property, which means that at certain points in time, the solution $\hat{L}(s, t)$, $\hat{P}(s, t)$ may not exist. This problem is known as the stabilizability problem in indirect APPC and further modifications are needed in order to handle it (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996). Assuming that the stabilizability condition holds at each time t, it can be shown (Goodwin and Sin 1984; Ioannou and Fidan 2006; Ioannou and Sun 1996) that all signals are bounded and the tracking error converges to zero with time. Other indirect adaptive pole placement control schemes include adaptive linear quadratic (Ioannou and Fidan 2006; Ioannou and Sun 1996). In principle any nonadaptive control scheme can be made adaptive by replacing the unknown parameters with their estimates in the calculation of the controller parameters. The design of direct APPC schemes is not possible in general as the map

between the plant and controller parameters is nonlinear, and the plant parameters cannot be expressed as a convenient function of the controller parameters. This prevents parametrization of the plant transfer function with respect to the controller parameters as done in the case of MRC. In special cases where such parametrization is possible such as in MRAC which can be viewed as a special case of APPC, the design of direct APPC is possible. Chapters on ▶ Adaptive Control, Overview, ▶ Robust Adaptive Control, and ▶ History of Adaptive Control provide additional information regarding MRAC and APPC.

Search Methods, Multiple Models, and Switching Schemes

One of the drawbacks of APPC is the stabilizability condition which requires the estimated plant at each time t to satisfy the detectability and stabilizability condition that is necessary for the controller parameters to exist. Since the adaptive law cannot guarantee such a property, an approach emerged that involves the pre-calculation of a set of controllers based on the partitioning of the plant parameter space. The problem then becomes one of identifying which one of the controllers is the most appropriate one. The switching to the "best" possible controller could be based on some logic that is driven by some cost index, multiple estimation models, and other techniques (Fekri et al. 2007; Hespanha et al. 2003; Kuipers and Ioannou 2010; Morse 1996; Narendra and Balakrishnan 1997; Stefanovic and Safonov 2011). One of the drawbacks of this approach is that it is difficult if at all possible to find a finite set of stabilizing controllers that cover the whole unknown parameter space especially for high-order plants. If found its dimension may be so large that makes it impractical. Another drawback that is present in all adaptive schemes is that in the absence of persistently exciting signals which guarantee that the input/output data have sufficient information about the unknown plant parameters, there is no guarantee that the controller the scheme converged to is indeed a stabilizing one. In other words, if switching is

disengaged or the adaptive law is switched off, there is no guarantee that a small disturbance will not drive the corresponding LTI scheme unstable. Nevertheless these techniques allow the incorporation of well-established robust control techniques in designing a priori the set of controller candidates. The problem is that if the plant parameters change in a way not accounted for a priori, no controller from the set may be stabilizing leading to an unstable system.

Robust Adaptive Control

The MRAC and APPC schemes presented above are designed for LTI systems. Due to the adaptive law, the closed-loop system is no longer LTI but nonlinear and time varying. It has been shown using simple examples that the pure integral action of the adaptive law could cause parameter drift in the presence of small disturbances and/or unmodeled dynamics (Ioannou and Fidan 2006; Ioannou and Kokotovic 1983; Ioannou and Sun 1996) which could then excite the unmodeled dynamics and lead to instability. Modifications to counteract these possible instabilities led to the field of robust adaptive control whose focus was to modify the adaptive law in order to guarantee robustness with respect to disturbances, unmodeled dynamics, time-varying parameters, classes of nonlinearities, etc., by using techniques such as normalizing signals, projection, fixed and switching sigma modification, etc.

Cross-References

▶ Adaptive Control, Overview
▶ History of Adaptive Control
▶ Model Reference Adaptive Control
▶ Robust Adaptive Control
▶ Switching Adaptive Control

Bibliography

Astrom K, Wittenmark B (1995) Adaptive control. Addison-Wesley, Reading
Egardt B (1979) Stability of adaptive controllers. Springer, New York

Fekri S, Athans M, Pascoal A (2007) Robust multiple model adaptive control (RMMAC): a case study. Int J Adapt Control Signal Process 21(1):1–30

Goodwin G, Sin K (1984) Adaptive filtering prediction and control. Prentice-Hall, Englewood Cliffs

Hespanha JP, Liberzon D, Morse A (2003) Hysteresis-based switching algorithms for supervisory control of uncertain systems. Automatica 39(2):263–272

Ioannou P, Fidan B (2006) Adaptive control tutorial. SIAM, Philadelphia

Ioannou P, Kokotovic P (1983) Adaptive systems with reduced models. Springer, Berlin/New York

Ioannou P, Sun J (1996) Robust adaptive control. Prentice-Hall, Upper Saddle River

Kuipers M, Ioannou P (2010) Multiple model adaptive control with mixing. IEEE Trans Autom Control 55(8):1822–1836

Landau Y (1979) Adaptive control: the model reference approach. Marcel Dekker, New York

Landau I, Lozano R, M'Saad M (1998) Adaptive control. Springer, New York

Morse A (1996) Supervisory control of families of linear set-point controllers part I: exact matching. IEEE Trans Autom Control 41(10):1413–1431

Narendra K, Annaswamy A (1989) Stable adaptive systems. Prentice Hall, Englewood Cliffs

Narendra K, Balakrishnan J (1997) Adaptive control using multiple models. IEEE Trans Autom Control 42(2):171–187

Sastry S, Bodson M (1989) Adaptive control: stability, convergence and robustness. Prentice Hall, Englewood Cliffs

Stefanovic M, Safonov M (2011) Safe adaptive control: data-driven stability analysis and robust synthesis. Lecture notes in control and information sciences, vol 405. Springer, Berlin

Tao G (2003) Adaptive control design and analysis. Wiley-Interscience, Hoboken

Adaptive Control, Overview

Richard Hume Middleton
School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Abstract

Adaptive control describes a range of techniques for altering control behavior using measured signals to achieve high control performance under uncertainty. The theory and practice of adaptive control has matured in many areas. This entry gives an overview of adaptive control with pointers to more detailed specific topics.

Keywords

Adaptive control; Estimation

Introduction

What Is Adaptive Control

Feedback control has a long history of using sensing, decision, and actuation elements to achieve an overall goal. The general structure of a control system may be illustrated in Fig. 1. It has long been known that high fidelity control relies on knowledge of the system to be controlled. For example, in most cases, knowledge of the plant gain and/or time constants (represented by θ_p in Fig. 1) is important in feedback control design. In addition, disturbance characteristics (e.g., frequency of a sinusoidal disturbance), θ_d in Fig. 1, are important in feedback compensator design.
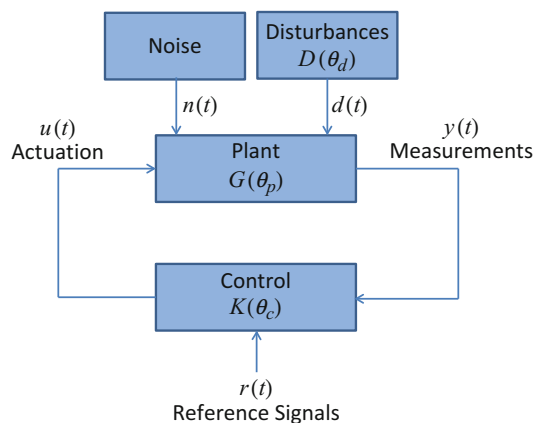
Many control design and synthesis techniques are model based, using prior knowledge of both model structure and parameters. In other cases, a fixed controller structure is used, and the controller parameters, θ_C in Fig. 1, are tuned empirically during control system commissioning.

Adaptive Control, Overview, Fig. 1 General control and adaptive control diagram

However, if the plant parameters vary widely with time or have large uncertainties, these approaches may be inadequate for high-performance control.

There are two main ways of approaching high-performance control with unknown plant and disturbance characteristics:

1. Robust control (▶ Optimization Based Robust Control), wherein a controller is designed to perform adequately despite the uncertainties. Variable structure control may have very high levels of robustness in some cases and therefore is a special class of robust nonlinear control.

2. Adaptive control, where the controller learns and adjusts its strategy based on measured data. This frequently takes the form where the controller parameters, θ_C, are time-varying functions that depend on the available data ($y(t)$, $u(t)$, and $r(t)$). Adaptive control has close links to intelligent control (including neural control (▶ Neural Control and Approximate Dynamic Programming), where specific types of learning are considered) and also to stochastic adaptive control (▶ Stochastic Adaptive Control).

Robust control is most useful when there are large unmodeled dynamics (i.e., structural uncertainties), relatively high levels of noise, or rapid and unpredictable parameter changes. Conversely, for slow or largely predictable parameter variations, with relatively well-known model structure and limited noise levels, adaptive control may provide a very useful tool for high-performance control (Åström and Wittenmark 2008).

Varieties of Adaptive Control

One practical variant of adaptive control is controller auto-tuning (▶ Autotuning). Auto-tuning is particularly useful for PID and similar controllers and involves a specific phase of signal injection, followed by analysis, PID gain computation, and implementation. These techniques are an important aid to commissioning and maintenance of distributed control systems.

There are also large classes of adaptive controllers that are continuously monitoring the plant input-output signals to adjust the strategy. These adjustments are often parametrized by a relatively small number of coefficients, θ_C. These include schemes where the controller parameters are directly adjusted using measureable data (also referred to as "implicit," since there is no explicit plant model generated). Early examples of this often included model reference adaptive control (▶ Model Reference Adaptive Control). Other schemes (Middleton et al. 1988) explicitly estimate a plant model θ_P; thereafter, performing online control design and, therefore, the adaptation of controller parameters θ_C are indirect. This then led on to a range of other adaptive control techniques applicable to linear systems (▶ Adaptive Control of Linear Time-Invariant Systems).

There have been significant questions concerning the sensitivity of some adaptive control algorithms to unmodeled dynamics, time-varying systems, and noise (Ioannou and Kokotovic 1984; Rohrs et al. 1985). This prompted a very active period of research to analyze and redesign adaptive control to provide suitable robustness (▶ Robust Adaptive Control) (e.g., Anderson et al. 1986; Ioannou and Sun 2012) and parameter tracking for time-varying systems (e.g., Kreisselmeier 1986; Middleton and Goodwin 1988).

Work in this area further spread to nonparametric methods, such as switching, or supervisory adaptive control (▶ Switching Adaptive Control) (e.g., Fu and Barmish 1986; Morse et al. 1992). In addition, there has been a great deal of work on the more difficult problem of adaptive control for nonlinear systems (▶ Nonlinear Adaptive Control).

A further adaptive control technique is extremum seeking control (▶ Extremum Seeking Control). In extremum seeking (or self optimizing) control, the desired reference for the system is unknown, instead we wish to maximize (or minimize) some variable in the system (Ariyur and Krstic 2003). These techniques have quite distinct modes of operation that have proven important in a range of applications.

A final control algorithm that has nonparametric features is iterative learning control (▶ Iterative Learning Control) (Amann et al. 1996; Moore 1993). This control scheme considers a system with a highly structured, namely, repetitive finite run, control problem. In this case, by taking a nonparametric approach of utilizing information from previous run(s), in many cases, near-perfect asymptotic tracking can be achieved.

Adaptive control has a rich history (▶ History of Adaptive Control) and has been established as an important tool for some classes of control problems.

Cross-References

Bibliography

Amann N, Owens DH, Rogers E (1996) Iterative learning control using optimal feedback and feedforward actions. Int J Control 65(2):277–293

Anderson BDO, Bitmead RR, Johnson CR, Kokotovic PV, Kosut RL, Mareels IMY, Praly L, Riedle BD (1986) Stability of adaptive systems: passivity and averaging analysis. MIT, Cambridge

Ariyur KB, Krstic M (2003) Real-time optimization by extremum-seeking control. Wiley, New Jersey

Åström KJ, Wittenmark B (2008) Adaptive control. Courier Dover Publications, Mineola

Fu M, Barmish BR (1986) Adaptive stabilization of linear systems via switching control. IEEE Trans Autom Control 31(12):1097–1103

Ioannou PA, Kokotovic PV (1984) Instability analysis and improvement of robustness of adaptive control. Automatica 20(5):583–594

Ioannou PA, Sun J (2012) Robust adaptive control. Dover Publications, Mineola/New York

Kreisselmeier G (1986) Adaptive control of a class of slowly time-varying plants. Syst Control Lett 8(2):97–103

Middleton RH, Goodwin GC (1988) Adaptive control of time-varying linear systems. IEEE Trans Autom Control 33(2):150–155

Middleton RH, Goodwin GC, Hill DJ, Mayne DQ (1988) Design issues in adaptive control. IEEE Trans Autom Control 33(1):50–58

Moore KL (1993) Iterative learning control for deterministic systems. Advances in industrial control series. Springer, London/New York

Morse AS, Mayne DQ, Goodwin GC (1992) Applications of hysteresis switching in parameter adaptive control. IEEE Trans Autom Control 37(9):1343–1354

Rohrs C, Valavani L, Athans M, Stein G (1985) Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. IEEE Trans Autom Control 30(9):881–889

Adaptive Cruise Control

Rajesh Rajamani
Department of Mechanical Engineering,
University of Minnesota, Twin Cities,
Minneapolis, MN, USA

Abstract

This chapter discusses advanced cruise control automotive technologies, including adaptive cruise control (ACC) in which spacing control, speed control, and a number of transitional maneuvers must be performed. The ACC system must satisfy difficult performance requirements of vehicle stability and string stability. The technical challenges involved and the control design techniques utilized in ACC system design are presented.

Keywords

Collision avoidance; String stability; Traffic stability; Vehicle following

Introduction

Adaptive cruise control (ACC) is an extension of cruise control. An ACC vehicle includes a radar, a lidar, or other sensor that measures the distance to any preceding vehicle in the same lane on the highway. In the absence of preceding vehicles, the speed of the car is controlled to a driver-desired value. In the presence of a preceding vehicle, the controller determines whether the vehicle should switch from speed control to spacing control. In spacing control, the distance to the preceding car is controlled to a desired value.

A different form of advanced cruise control is a *forward collision avoidance* (FCA) system. An FCA system uses a distance sensor to determine if the vehicle is approaching a car ahead too quickly and will automatically apply brakes to minimize the chances of a forward collision. For the 2013 model year, 29 % vehicles have forward collision warning as an available option and 12 % include autonomous braking for a full FCA system. Examples of models in which an FCA system is standard are the Mercedes Benz G-class and the Volvo S-60, S-80, XC-60, and XC-70.

It should be noted that an FCA system does not involve steady-state vehicle following. An ACC system on the other hand involves control of speed and spacing to desired steady-state values.

ACC systems have been in the market in Japan since 1995, in Europe since 1998, and in the US since 2000. An ACC system provides enhanced driver comfort and convenience by allowing extended operation of the cruise control option even in the presence of other traffic.

Controller Architecture

The ACC system has two modes of steady state operation: speed control and vehicle following (i.e., spacing control). Speed control is traditional cruise control and is a well-established technology. A proportional-integral controller based on feedback of vehicle speed (calculated from rotational wheel speeds) is used in cruise control (Rajamani 2012).
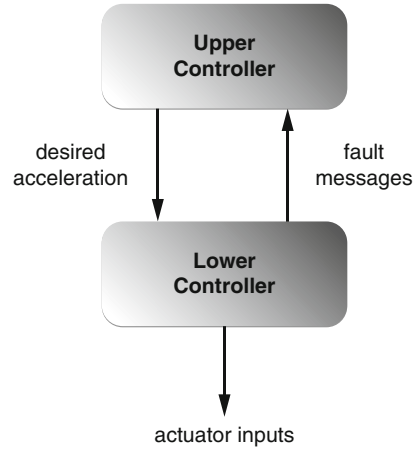
Adaptive Cruise Control, Fig. 1 Structure of longitudinal control system

Controller design for vehicle following is the primary topic of discussion in the sections titled "Vehicle Following Requirements" and "String Stability Analysis" in this chapter.

Transitional maneuvers and transitional control algorithms are discussed in the section titled "Transitional Maneuvers" in this chapter.

The longitudinal control system architecture for an ACC vehicle is typically designed to be hierarchical, with an upper-level controller and a lower-level controller, as shown in Fig. 1.

The upper-level controller determines the desired acceleration for the vehicle. The lower level controller determines the throttle and/or brake commands required to track the desired acceleration. Vehicle dynamic models, engine maps, and nonlinear control synthesis techniques are used in the design of the lower controller (Rajamani 2012). This chapter will focus only on the design of the upper controller, also known as the ACC controller.

As far as the upper-level controller is concerned, the plant model for control design is

$$\ddot{x}_i = u \tag{1}$$

where the subscript i denotes the ith car in a string of consecutive ACC cars. The acceleration of the car is thus assumed to be the control input. However, due to the finite bandwidth associated

Adaptive Cruise Control,
Fig. 2 String of adaptive
cruise control vehicles
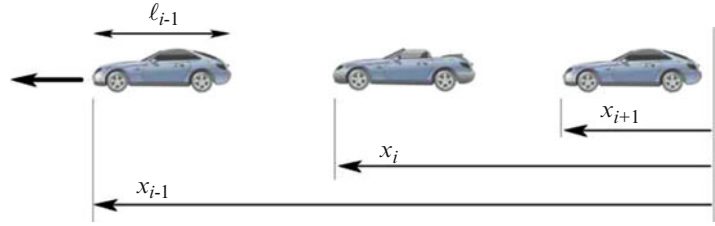
with the lower level controller, each car is actually expected to track its desired acceleration imperfectly. The objective of the upper level controller design is therefore stated as that of meeting required performance specifications robustly in the presence of a first order lag in the lower-level controller performance:

$$\ddot{x}_i = \frac{1}{\tau s + 1}\ddot{x}_{i\_\text{des}} = \frac{1}{\tau s + 1}u_i. \qquad (2)$$

Equation (1) is thus assumed to be the nominal plant model while the performance specifications have to be met even if the actual plant model were given by Eq. (2). The lag τ typically has a value between 0.2 and 0.5 s (Rajamani 2012).

Vehicle Following Requirements

In the vehicle following mode of operation, the ACC vehicle maintains a desired spacing from the preceding vehicle. The two important performance specifications that the vehicle following control system must satisfy are: individual vehicle stability and string stability.

(a) **Individual vehicle stability**
Consider a string of vehicles on the highway using a longitudinal control system for vehicle following, as shown in Fig. 2. Let x_i be the location of the ith vehicle measured from an inertial reference. The spacing error for the ith vehicle (the ACC vehicle under consideration) is then defined as

$$\delta_i = x_i - x_{i-1} + L_\text{des}. \qquad (3)$$

Here, L_des is the desired spacing and includes the preceding vehicle length ℓ_{i-1}. L_des could be chosen as a function of variables such as the

vehicle speed $\dot{x}_i$. The ACC control law is said to provide individual vehicle stability if the spacing error of the ACC vehicle converges to zero when the preceding vehicle is operating at constant speed:

$$\ddot{x}_{i-1} \to 0 \Rightarrow \delta_i \to 0. \qquad (4)$$

(b) **String stability**
The spacing error is expected to be non-zero during acceleration or deceleration of the preceding vehicle. It is important then to describe how the spacing error would propagate from vehicle to vehicle in a string of ACC vehicles during acceleration. The string stability of a string of ACC vehicles refers to a property in which spacing errors are guaranteed not to amplify as they propagate towards the tail of the string (Swaroop and Hedrick 1996).

String Stability Analysis

In this section, mathematical conditions that ensure string stability are provided.

Let δ_i and δ_{i-1} be the spacing errors of consecutive ACC vehicles in a string. Let $\hat{H}(s)$ be the transfer function relating these errors:

$$\hat{H}(s) = \frac{\hat{\delta}_i}{\hat{\delta}_{i-1}}(s). \qquad (5)$$

The following two conditions can be used to determine if the system is string stable:
(a) The transfer function $\hat{H}(s)$ should satisfy

$$\left\| \hat{H}(s) \right\|_\infty \le 1. \qquad (6)$$

(b) The impulse response function $h(t)$ corresponding to $\hat{H}(s)$ should not change sign (Swaroop and Hedrick 1996), i.e.,

$$h(t) > 0 \quad \forall t \geq 0. \tag{7}$$

The reasons for these two requirements to be satisfied are described in Rajamani (2012). Roughly speaking, Eq. (6) ensures that $||\delta_i||_2 \leq ||\delta_{i-1}||_2$, which means that the energy in the spacing error signal decreases as the spacing error propagates towards the tail of the string. Equation (7) ensures that the steady state spacing errors of the vehicles in the string have the same sign. This is important because a positive spacing error implies that a vehicle is closer than desired while a negative spacing error implies that it is further apart than desired. If the steady state value of δ_i is positive while that of δ_{i-1} is negative, then this might be dangerous due to the vehicle being closer, even though in terms of magnitude δ_i might be smaller than δ_{i-1}.

If conditions (6) and (7) are both satisfied, then $||\delta_i||_\infty \leq ||\delta_{i-1}||_\infty$ (Rajamani 2012).

Constant Inter-vehicle Spacing

The ACC system only utilizes on board sensors like radar and does not depend on inter-vehicle communication from other vehicles. Hence the only variables available as feedback for the upper controller are inter-vehicle spacing, relative velocity and the ACC vehicle's own velocity.

Under the constant spacing policy, the spacing error of the ith vehicle was defined in Eq. (3).

If the acceleration of the vehicle can be instantaneously controlled, then it can be shown that a linear control system of the type

$$\ddot{x}_i = -k_p \delta_i - k_v \dot{\delta}_i \tag{8}$$

results in the following closed-loop transfer function between consecutive spacing errors

$$\hat{H}(s) = \frac{\hat{\delta}_i}{\hat{\delta}_{i-1}}(s) = \frac{k_p + k_v s}{s^2 + k_v s + k_p}. \tag{9}$$

Equation (9) describes the propagation of spacing errors along the vehicle string.

All positive values of k_p and k_v guarantee individual vehicle stability. However, it can be shown that there are no positive values of k_p and k_v for which the magnitude of $G(s)$ can be guaranteed to be less than unity at all frequencies. The details of this proof are available in Rajamani (2012).

Thus, the constant spacing policy will always be string unstable.

Constant Time-Gap Spacing

Since the constant spacing policy is unsuitable for autonomous control, a better spacing policy that can ensure both individual vehicle stability and string stability must be used. The constant time-gap (CTG) spacing policy is such a spacing policy. In the CTG spacing policy, the desired inter-vehicle spacing is not constant but varies with velocity. The spacing error is defined as

$$\delta_i = x_i - x_{i-1} + L_{\text{des}} + h\dot{x}_i. \tag{10}$$

The parameter h is referred to as the time-gap.

The following controller based on the CTG spacing policy can be used to regulate the spacing error at zero (Swaroop et al. 1994):

$$\ddot{x}_{i\_\text{des}} = -\frac{1}{h}(\dot{x}_i - \dot{x}_{i-1} + \lambda \delta_i) \tag{11}$$

With this control law, it can be shown that the spacing errors of successive vehicles δ_i and δ_{i-1} are independent of each other:

$$\dot{\delta}_i = -\lambda \delta_i \tag{12}$$

Thus, δ_i is independent of δ_{i-1} and is expected to converge to zero as long as $\lambda > 0$. However, this result is only true if any desired acceleration can be instantaneously obtained by the vehicle i.e., if $\tau = 0$.

In the presence of the lower controller and actuator dynamics given by Eq. (2), it can be shown that the dynamic relation between δ_i and δ_{i-1} in the transfer function domain is

$$\hat{H}(s) = \frac{s + \lambda}{h\tau s^3 + hs^2 + (1 + \lambda h)s + \lambda} \quad (13)$$

The string stability of this system can be analyzed by checking if the magnitude of the above transfer function is always less than or equal to 1. It can be shown that this is the case at all frequencies if and only if (Rajamani 2012)

$$h \geq 2\tau. \quad (14)$$

Further, if Eq. (14) is satisfied, then it is also guaranteed that one can find a value of λ such that Eq. (7) is satisfied. Thus the condition (14) is necessary (Swaroop and Hedrick 1996) for string stability.

Since the typical value of τ is of the order of 0.5 s, Eq. (14) implies that ACC vehicles must maintain at least a 1-s time gap between vehicles for string stability.

Transitional Maneuvers

While under speed control, an ACC vehicle might suddenly encounter a new vehicle in its lane (either due to a lane change or due to a slower moving preceding vehicle). The ACC vehicle must then decide whether to continue to operate under the speed control mode or transition to the vehicle following mode or initiate hard braking. If a transition to vehicle following is required, a transitional trajectory that will bring the ACC vehicle to its steady state following distance needs to be designed. Similarly, a decision on the mode of operation and design of a transitional trajectory are required when an ACC vehicle loses its target.

The regular CTG control law cannot directly be used to follow a newly encountered vehicle, see Rajamani (2012) for illustrative examples.

When a new target vehicle is encountered by the ACC vehicle, a "range – range rate" diagram can be used (Fancher and Bareket 1994) to decide if

(a) The vehicle should use speed control.
(b) The vehicle should use spacing control (with a defined transition trajectory in which desired spacing varies slowly with time)
(c) The vehicle should brake as hard as possible in order to avoid a crash.

The maximum allowable values for acceleration and deceleration need to be taken into account in making these decisions.

For the range – range rate $(R - \dot{R})$ diagram, define range R and range rate $\dot{R}$ as
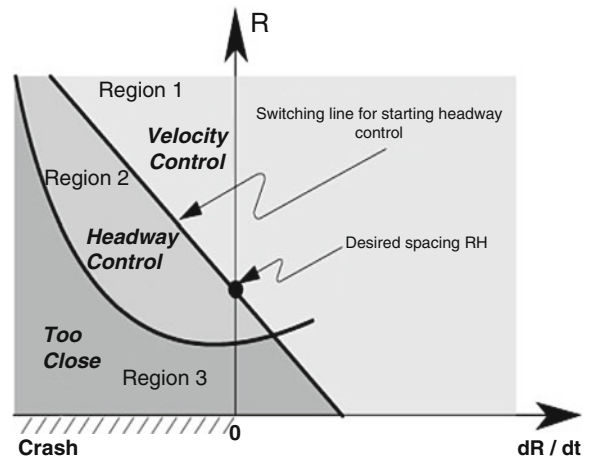
$$R = x_{i-1} - x_i \quad (15)$$

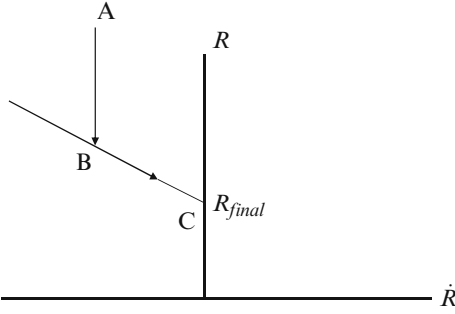$$\dot{R} = \dot{x}_{i-1} - \dot{x}_i = V_{i-1} - V_i \quad (16)$$

where x_{i-1}, x_i, V_{i-1}, and V_i are inertial positions and velocities of the preceding vehicle and the ACC vehicle respectively.

A typical $R - \dot{R}$ diagram is shown in Fig. 3 (Fancher and Bareket 1994). Depending on the

Adaptive Cruise Control,
Fig. 3 Range vs. range-rate diagram

Adaptive Cruise Control, Fig. 4 Switching line for spacing control

measured real-time values of R and $\dot{R}$, and the $R - \dot{R}$ diagram in Fig. 3, the ACC system determines the mode of longitudinal control. For instance, in region 1, the vehicle continues to operate under speed control. In region 2, the vehicle operates under spacing control. In region 3, the vehicle decelerates at the maximum allowed deceleration so as to try and avoid a crash.

The switching line from speed to spacing control is given by

$$R = -T\dot{R} + R_{\text{final}} \qquad (17)$$

where T is the slope of the switching line. When a slower vehicle is encountered at a distance larger than the desired final distance R_{final}, the switching line shown in Fig. 4 can be used to determine when and whether the vehicle should switch to spacing control. If the distance R is greater than that given by the line, speed control should be used.

The overall strategy (shown by trajectory ABC) is to first reduce gap at constant $\dot{R}$ and then follow the desired spacing given by the switching line of Eq. (17).

The control law during spacing control on this transitional trajectory is as follows. Depending on the value of $\dot{R}$, determine R from Eq. (17). Then use R as the desired inter-vehicle spacing in the PD control law

$$\ddot{x}_{\text{des}} = -k_p \left(x_i - R\right) - k_d \left(\dot{x}_i - \dot{R}\right) \qquad (18)$$

The trajectory of the ACC vehicle during constant deceleration is a parabola on the $R - \dot{R}$ diagram (Rajamani 2012).

The switching line should be such that travel along the line is comfortable and does not constitute high deceleration. The deceleration during coasting (zero throttle and zero braking) can be used to determine the slope of the switching line (Rajamani 2012).

Note that string stability is not a concern during transitional maneuvers (Rajamani 2012).

Traffic Stability

In addition to individual vehicle stability and string stability, another type of stability analysis that has received significant interest in ACC literature is traffic flow stability. Traffic flow stability refers to the stable evolution of traffic velocity and traffic density on a highway section, for given inflow and outflow conditions. One well-known result in this regard in literature is that traffic flow is defined to be stable if $\frac{\partial q}{\partial \rho}$ is positive, i.e., as the density ρ of traffic increases, traffic flow rate q must increase (Swaroop and Rajagopal 1999). If this condition is not satisfied, the highway section would be unable to accommodate any constant inflow of vehicles from an oncoming ramp. The steady state traffic flow on the highway section would come to a stop, if the ramp inflow did not stop (Swaroop and Rajagopal 1999).

It has been shown that the constant time-gap spacing policy used in ACC systems has a negative $q - \rho$ slope and thus does not lead to traffic flow stability (Swaroop and Rajagopal 1999). It has also been shown that it is possible to design other spacing policies (in which the desired spacing between vehicles is a nonlinear function of speed, instead of being proportional to speed) that can provide stable traffic flow (Santhanakrishnan and Rajamani 2003).

The importance of traffic flow stability has not been fully understood by the research community. Traffic flow stability is likely to become important when the number of ACC vehicles

on the highway increase and their penetration percentage into vehicles on the road becomes significant.

Recent Automotive Market Developments

The latest versions of ACC systems on the market have been enhanced with collision warning, integrated brake support, and stop-and-go operation functionality.

The collision warning feature uses the same radar as the ACC system to detect moving vehicles ahead and determine whether driver intervention is required. In this case, visual and audio warnings are provided to alert the driver and brakes are pre-charged to allow quick deceleration. On Ford's ACC-equipped vehicles, brakes are also automatically applied when the driver lifts the foot off from the accelerator pedal in a detected collision warning scenario.

When enabled with stop-and-go functionality, the ACC system can also operate at low vehicle speeds in heavy traffic. The vehicle can be automatically brought to a complete stop when needed and restarted automatically. Stop-and-go is an expensive option and requires the use of multiple radar sensors on each car. For instance, the BMW ACC system uses two short range and one long range radar sensor for stop-and-go operation.

The 2013 versions of ACC on the Cadillac ATS and on the Mercedes Distronic systems are also being integrated with camera based lateral lane position measurement systems. On the Mercedes Distronic systems, a camera steering assist system provides automatic steering, while on the Cadillac ATS, a camera based system provides lane departure warnings.

Future Directions

Current ACC systems use only on-board sensors and do not use wireless communication with other vehicles. There is a likelihood of evolution of current systems into co-operative adaptive cruise control (CACC) systems which utilize wireless communication with other vehicles and highway infrastructure. This evolution could be facilitated by the dedicated short-range communications (DSRC) capability being developed by government agencies in the US, Europe and Japan. In the US, DSRC is being developed with a primary goal of enabling communication between vehicles and with infrastructure to reduce collisions and support other safety applications. In CACC, wireless communication could provide acceleration signals from several preceding downstream vehicles. These signals could be used in better spacing policies and control algorithms to improve safety, ensure string stability, and improve traffic flow.

Cross-References

► Lane Keeping
► Vehicle Dynamics Control
► Vehicular Chains

Bibliography

Fancher P, Bareket Z (1994) Evaluating headway control using range versus range-rate relationships. Veh Syst Dyn 23(8):575–596

Rajamani R (2012) Vehicle dynamics and control, 2nd edn. Springer, New York. ISBN:978-1461414322

Santhanakrishnan K, Rajamani R (2003) On spacing policies for highway vehicle automation. IEEE Trans Intell Transp Syst 4(4):198–204

Swaroop D, Hedrick JK (1996) String stability of interconnected dynamic systems. IEEE Trans Autom Control 41(3):349–357

Swaroop D, Rajagopal KR (1999) Intelligent cruise control systems and traffic flow stability. Transp Res C Emerg Technol 7(6):329–352

Swaroop D, Hedrick JK, Chien CC, Ioannou P (1994) A comparison of spacing and headway control laws for automatically controlled vehicles. Veh Syst Dyn 23(8):597–625

Advanced Manipulation for Underwater Sampling

Giuseppe Casalino
University of Genoa, Genoa, Italy

Abstract

This entry deals with the kinematic self-coordination aspects to be managed by parts of underwater floating manipulators, whenever employed for sample collections at the seafloor.

Kinematic self-coordination is here intended as the autonomous ability exhibited by the system in closed loop specifying the most appropriate reference velocities for its main constitutive parts (i.e., the supporting vehicle and the arm) in order to execute the sample collection with respect to both safety and best operability conditions for the system while also guaranteeing the needed "execution agility" in performing the task, particularly useful in case of underwater repeated collections. To this end, the devising and employment of a unifying control framework capable of guaranteeing the above properties will be outlined.

Such a framework is however intended to only represent the so-called Kinematic Control Layer (KCL) overlaying a Dynamic Control Layer (DCL), where the overall system dynamic and hydrodynamic effects are suitably accounted for, to the benefit of closed loop tracking of the reference system velocities. Since the DCL design is carried out in a way which is substantially independent from the system mission(s), it will not constitute a specific topic of this entry, even if some orienting references about it will be provided.

At this entry's end, as a follow-up of the resulting structural invariance of the devised KCL framework, future challenges addressing much wider and complex underwater applications will be foreseen, beyond the here-considered sample collection one.

Keywords

Introduction

An automated system for underwater sampling is here intended to be an autonomous underwater floating manipulator (see Fig. 1) capable of collecting samples corresponding to an a priori assigned template. The snapshots of Fig. 1 outline the most recent realization of a system of this kind (completed in 2012 within the EU-funded project TRIDENT; Sanz et al. 2012) when in operation, which is characterized by a vehicle and an endowed 7-dof arm exhibiting comparable masses and inertia, thus resulting in potentially faster and more agile designs than the very few similar previous realizations.

Its general the operational mode consists in exploring an assigned area of the seafloor, while executing a collection each time a feature corresponding to the assigned template is recognized (by the vehicle endowed with a stereovision system) as a sample to be collected.

Thus the autonomous functionalities to be exhibited are the following (to be sequenced as they are listed on an event-driven basis): (1) explore an assigned seabed area while visually performing model-based sample recognitions, (2) suspend the exploration and grasping a recognized sample, (3) deposit the sample inside an endowed container, and (4) then restart exploring till the next recognized sample.

Functionalities (1) and (4), since they do not require the arm usage, naturally reenter within the topics of navigation, patrolling, visual mapping, etc., which are typical of traditional AUVs and consequently will not be discussed here. Only functionality (2) will be discussed, since it is most distinctive of the considered system (often termed as I-AUV, with "I" for "Intervention") and because functionality (3) can be established along the same lines of (2) as a particular simpler case.
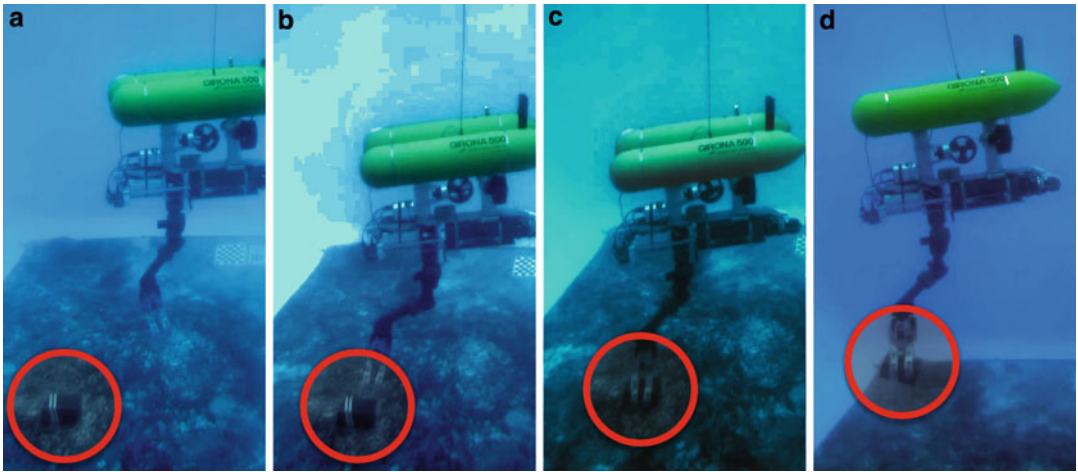
Advanced Manipulation for Underwater Sampling, Fig. 1 Snapshots showing the underwater floating manipulator TRIDENT when autonomously picking an identified object

By then focusing on functionality (2), we must note how the sample grasping ultimate objective, which translates into a specific position/attitude to be reached by the end-effector, must however be achieved within the preliminary fulfillment of also other objectives, each one reflecting the need of guaranteeing the system operating within both its safety and best operability conditions. For instance, the arm's joint limits must be respected and the arm singular postures avoided. Moreover, since the sample position is estimated via the vehicle with a stereo camera, the sample must stay grossly centered inside its visual cone, since otherwise the visual feedback would be lost and the sample search would need to start again. Also, the sample must stay within suitable horizontal and vertical distance limits from the camera frame, in order for the vision algorithm to be well performing. And furthermore, in these conditions the vehicle should be maintained with an approximately horizontal attitude, for energy savings.

With the exception of the objective of making the end-effector position/attitude reaching the grasping position, which is clearly an equality condition, its related safety/enabling objectives are instead represented by a set of inequality conditions (involving various system variables) whose achievement (accordingly with their safety/enabling role) must therefore deserve the highest priority.

System motions guaranteeing such prioritized objective achievements should moreover allow for a concurrent management of them (i.e., avoiding a sequential motion management whenever possible), which means requiring each objective progressing toward its achievement, by at each time instant only exploiting the residual system mobility allowed by the current progresses of its higher priority objectives. Since the available system mobility will progressively increase during time, accordingly with the progressive achievement of all inequality objectives, this will guarantee the grasping objective to be also completed by eventually progressing within adequate system safety and best operability conditions. In this way the system will also exhibit the necessary "agility" in executing its maneuvers, in a way faster than in case they were executed on a sequential motion basis.

The devising of an effective way to incorporate all the inequality and equality objectives within a uniform and computationally efficient task-priority-based algorithmic framework for underwater floating manipulators has been the

result of the developments outlined in the next section.

The developed framework however solely represents the so-called Kinematic Control Layer (KCL) of the overall control architecture, that is, the one in charge of closed-loop real-time control generating the system velocity vector y as a reference signal, to be in turn concurrently tracked, via the action of the arm joint torques and vehicle thrusters, by an adequate underlying Dynamic Control Layer (DCL), where the overall dynamic and hydrodynamic effects are kept into account to the benefit of such velocity tracking. Since the DCL can actually be designed in a way substantially independent from the system mission(s), it will not constitute a specific topic of this entry. Its detailed dynamic-hydrodynamic model-based structuring, also including a stability analysis, can be found in Casalino (2011), together with a more detailed description of the upper-lying KCL, while more general references on underwater dynamic control aspects can be found, for instance, in Antonelli (2006).

Task-Priority-Based Control of Floating Manipulators

The above-outlined typical set of objectives (of inequality and/or equality types) to be achieved within a sampling mission are here formalized. Then some helpful generalizing definitions are given, prior to presenting the related unifying task-priority-based algorithmic framework to be used.

Inequality and Equality Objectives

One of the objectives, of inequality type, related to both arm safety and its operability is that of maintaining each joint within corresponding minimum and maximum limits, that is,

$$q_{1m} < q_i < q_{iM}; \quad i = 1, 2, \ldots, 7$$

Moreover, in order to have the arm operating with dexterity, its manipulability measure (Nakamura 1991; Yoshikawa 1985) must ultimately stay above a minimum threshold value, thus also

requiring the achievement of the inequality type objective

$$\mu > \mu_m$$

While the above objectives arise from inherently scalar variables, other objectives instead arise as conditions to be achieved within the Cartesian space, where each one of them can be conveniently expressed in terms of the modulus associated to a corresponding Cartesian vector variable.

To be more specific, let us, for instance, refer to the need of avoiding the occlusions between the sample and the stereo camera, which might occasionally occur due to the arm link motions. Then such need can be, for instance, translated into the ultimate achievement of the following set of inequalities, for suitable chosen values of the boundaries

$$\|l\| > l_m; \quad \|\tau\| > \tau_m; \quad \|\eta\| < \eta_M$$

where l is the vector lying on the vehicle x-y plane, joining the arm elbow with the line parallel to the vehicle z-axis and passing through camera frame origin, as sketched in Fig. 2a. Moreover η is the misalignment vector formed by vector τ also lying on the vehicle x-y plane, joining the lines parallel to the vehicle z-axis and, respectively, passing through the elbow and the end-effector origin.

As for the vehicle, it must keep the object of interest grossly centered in the camera frame (see Fig. 2b), thus meaning that the modulus of the orientation error ξ, formed by the unit vector n_p of vector p from the sample to the camera frame and the unit vector k_c of the z-axis of the camera frame itself, must ultimately satisfy the inequality

$$\|\xi\| < \xi_M$$

Furthermore, the camera must also be closer than a given horizontal distance d_M to the vertical line passing through the sample, and it must lie between a maximum and minimum height with respect to the sample itself, thus implying the achievement of the following inequalities (Fig. 2c, d):
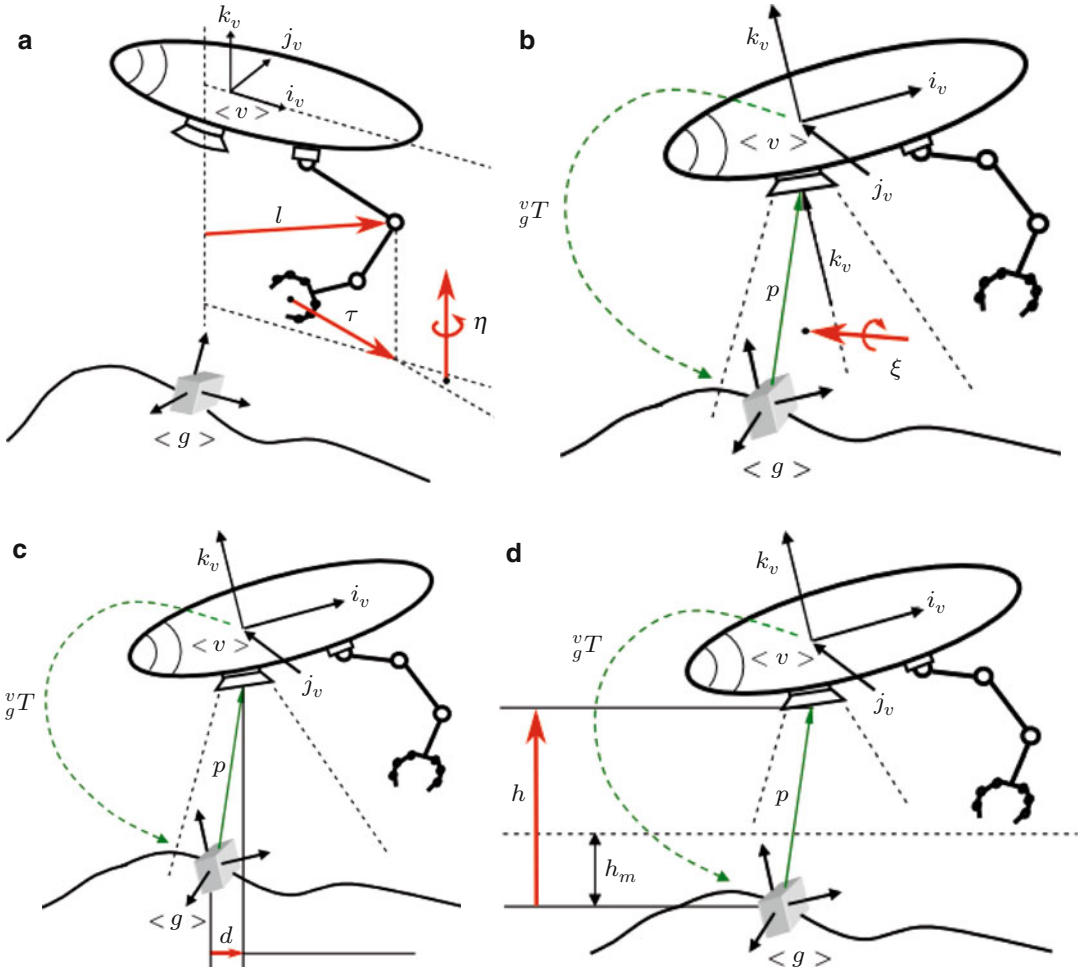
Advanced Manipulation for Underwater Sampling, Fig. 2 Vectors allowing for the defintion of some inequality objectives in the Cartesian space: (*a*) camera occlusion, (*b*) camera centering, (*c*) camera distance, (*d*) camera height

$$\|d\| < d_M; \quad h_m < \|h\| < h_M$$

Also since the vehicle should exhibit an almost horizontal attitude, this further requires the achievement of the following additional inequality:

$$\|\phi\| < \phi_M$$

with ϕ the misalignment vector formed by the absolute vertical unit vector k_o with the vehicle z-axis one k_v.

And finally the end-effector must eventually reach the sample, for then picking it. Thus the following, now of equality type, objectives must also be ultimately achieved, where r is the position error and θ the orientation one of the end-effector frame with respect to the sample frame

$$\|r\| = 0; \quad \|\vartheta\| = 0$$

As already repeatedly remarked, the achievement of the above inequality objectives (since related to the system safety and/or its best operability) must globally deserve a priority higher than the last equality.

Basic Definitions

The following definitions only regard a generic vector $s \in R^3$ characterizing a corresponding generic objective defined in the Cartesian space

(for instance, with the exclusion of the joint and manipulability limits, all the other above-reported objectives). In this case the vector is termed to be the error vector of the objective, and it is assumed measured with components on the vehicle frame. Then its modulus

$$\sigma \doteq \|s\|$$

is termed to be the error, while its unit vector

$$n \doteq s/\sigma; \quad \sigma \neq 0$$

is accordingly denoted as the unit error vector. Then the following differential Jacobian relationship can always be evaluated for each of them:

$$\dot{s} = Hy$$

where $y \in R^N$ ($N = (7 + 6)$ for the system of Fig. 1) is the stacked vector composed of the joint velocity vector $\dot{q} \in R^7$, plus the stacked vector $v \in R^6$ of the absolute vehicle velocities (linear and angular) with components on the vehicle frame and with $\dot{s}$ clearly representing the time derivative of vector s itself, as seen from the vehicle frame and with components on it (see Casalino (2011) for details on the real-time evaluation of Jacobian matrices H).

Obviously, for the time derivative $\dot{\sigma}$ of the error, also the following differential relationship holds

$$\dot{\sigma} = n^T Hy$$

Further, to each error variable σ, a so-called error reference rate is real time assigned of the form

$$\dot{\bar{\sigma}} = -\gamma(\sigma - \sigma^o)\alpha(\sigma)$$

where for equality objectives σ^o is the target value and $\alpha(\sigma) \equiv 1$, while for inequality ones, σ^o is the threshold value and $\alpha(\sigma)$ is a left-cutting or right-cutting (in correspondence of σ^o) smooth sigmoidal activation function, depending on whether the objective is to force σ to be below or above σ^o, respectively.

In case $\dot{\bar{\sigma}}$ could be exactly assigned to its corresponding error rate $\dot{\sigma}$, it would consequently smoothly drive σ toward the achievement of its associated objective. Note however that for inequality objectives, it would necessarily impose $\dot{\sigma} = 0$ in correspondence of a point located inside the interval of validity of the inequality objective itself, while instead such an error rate zeroing effect should be relaxed, for allowing the helpful subsequent system mobility increase, which allows for further progress toward other lower priority control objectives. Such a relaxation aspect will be dealt with soon.

Furthermore, in correspondence of a reference error rate $\dot{\bar{\sigma}}$, the so-called reference error vector rate can also be defined as

$$\dot{\bar{s}} \doteq n\dot{\bar{\sigma}}$$

that for equality objectives requiring the zeroing of their error σ simply becomes

$$\dot{\bar{s}} \doteq -\gamma s$$

whose evaluation, since not requiring its unit vector n, will be useful for managing equality objectives.

Finally note that for each objective not defined in the Cartesian space (like, for instance, the above joint limits and manipulability), the corresponding scalar error variable, its rate, and its reference error rate can instead be managed directly, since obviously they do not require any preliminary scalar reduction process.

Managing the Higher Priority Inequality Objectives

A prioritized list of the various scalar inequality objectives, to be concurrently progressively achieved, is suitably established in a descending priority order.

Then, by starting to consider the highest priority one, we have that the linear manifold of the system velocity vector y (i.e., the arm joints velocity vector $\dot{q}$ stacked with vector v of the vehicle linear and angular velocities), capable of driving toward its achievement, results at each

time instant as the set of solution of the following minimization problem with scalar argument, with row vector $G_1 \doteq \alpha_1 n_1^T H_1$ and scalar α_1 the same activation function embedded within the reference error rate $\dot{\bar{\sigma}}_1$

$$
S_1 \doteq \left\{ \underset{y}{\mathrm{argmin}} \left\| \dot{\bar{\sigma}}_1 - G_1 y \right\|^2 \right\} \quad \Leftrightarrow
$$
$$
y = G_1^{\#} \dot{\bar{\sigma}}_1 + (I - G_1^{\#} G_1) z_1 \doteq \rho_1 + Q_1 z_1; \ \forall z_1
$$
$$(1)$$

The above minimization, whose solution manifold appears at the right (also expressed in a concise notation with an obvious correspondence of terms) parameterized by the arbitrary vector z_1, has to be assumed executed without extracting the common factor α_1, that is, by evaluating the pseudo-inverse matrix $G_1^{\#}$ via the regularized form

$$
G_1^{\#} = \left(\alpha_1^2 n_1^T H_1 H_1^T n_1 + p_1 \right)^{-1} \alpha_1 H_1^T n_1
$$

with p_1, a suitably chosen bell-shaped, finite support and centered on zero, regularizing function of the norm of row vector G_1.

In the above solution manifold, when $\alpha_1 = 1$ (i.e., when the first inequality is still far to be achieved), the second arbitrary term $Q_1 z_1$ is orthogonal to the first, thus having no influence on the generated $\dot{\sigma}_1 = \dot{\bar{\sigma}}_1$ and consequently suitable to be used for also progressing toward the achievement of other lower priority objectives, without perturbing the current progressive achievement of the first one. Note however that, since in this condition the span of the second term results one dimension less than the whole system velocity space $y \in R^N$, this implies that the lower priority objectives can be progressed by only acting within a one-dimension reduced system velocity subspace.

When $\alpha_1 = 0$ (i.e., when the first inequality is achieved) since $G_1^{\#} = 0$ (as granted by the regularization) and consequently $y = z_1$, the lower priority objectives can instead be progressed by now exploiting the whole system velocity space.

When instead α_1 is within its transition zone $0 < \alpha_1 < 1$ (i.e., when the first inequality is near to be achieved), since the two terms of the solution manifold now become only approximately orthogonal, this can make the usage of the second term for managing lower priority tasks, possibly counteracting the first, currently acting in favor of the highest priority one, but in any case without any possibility of making the primary error variable σ_1 getting out of its enlarged boundaries (i.e., the ones inclusive of the transition zone), thus meaning that once the primary variable σ_1 has entered within such larger boundaries, it will definitely never get out of them.

With the above considerations in mind, managing the remaining priority-descending sequence of inequality objectives can then be done by applying the same philosophy to each of them and within the mobility space left free by its preceding ones, that is, as the result of the following sequence of nested minimization problems:

$$
S_i \doteq \left\{ \underset{y \in S_{i-1}}{\mathrm{argmin}} \left\| \dot{\bar{\sigma}}_i - G_i y \right\|^2 \right\}; \quad i = 1, 2, \ldots, k
$$

with $G_i \doteq \alpha_i n_i^T H_i$ and with k indexing the lowest priority inequality objective and where the highest priority objective has been also included for the sake of completeness (upon letting $S_o = R^N$). In this way the procedure guarantees the concurrent prioritized convergence (actually occurring as a sort of "domino effect" scattering along the prioritized objective list) toward the ultimate fulfillment of all inequality objectives, each one within its enlarged bounds at worse and with no possibility of getting out of them, once reached.

Further, a simple algebra allows translating the above sequence of k nested minimizations into the following algorithmic structure, with initialization $\rho_0 = 0$; $Q_0 = I$ (see Casalino et al. 2012a,b for more details):

$$
\hat{G}_1 \doteq G_i Q_i
$$
$$
T_i = \left(I - Q_{i-1} G_i^{\#} G_i \right)
$$

$$\rho_i = T_i \rho_{i-1} + Q_{i-1} G_i^{\#} \dot{\sigma}_i$$

$$Q_i = Q_{i-1} \left(I - G_i^{\#} G_i \right)$$

ending with the last k-th iteration with the solution manifold

$$y = \rho_k + Q_k z_k; \quad \forall z_k$$

where the residual arbitrariness space $Q_k z_z$ has to be then used for managing the remaining equality objectives, as hereafter indicated.

Managing the Lower Priority Equality Objectives and Subsystem Motion Priorities

For managing the lower priority equality objectives when these require the zeroing of their associated error σ_i (as, for instance, for the end-effector sample reaching task), the following sequence of nested minimization problems has to be instead considered (with initialization ρ_k; Q_k):

$$S_i \doteq \left\{ \underset{y \in S_{i-1}}{\operatorname{argmin}} \left\| \dot{\bar{s}}_i - H_i y \right\|^2 \right\}; \ i = (k+1), \ldots, m$$

with m indexing the last priority equality objective and where the whole reference error vector rates $\dot{\bar{s}}_i$ and associated whole error vectors $\dot{s}_i$ have now to be used, since for $\alpha_i \equiv 1$ (as it is for any equality objective) the otherwise needed evaluation of unit vectors n_i (which become ill defined for the relevant error σ_i approaching zero) would most probably provoke unwanted chattering phenomena around $\sigma_i = 0$, while instead the above avoids such risk (since $\dot{\bar{s}}_i$ and $\dot{s}_i$ can be evaluated without requiring n_i), even if at the cost of requiring, for each equality objective, three degrees of mobility instead of a sole one, as it instead is for each inequality objectives. However, note how the algorithmic translation of the above procedure remains structurally the same as the one for the inequality objectives (obviously with the substitutions $\dot{\bar{s}}_i \rightarrow \dot{\sigma}_i$, $H_i \rightarrow G_i$, and with initialization ρ_k, Q_k), thus ending in correspondence of the m-th last equality objective with the solution manifold

$$y = \rho_m + Q_m z_m; \quad \forall z_m$$

where the still possibly existing residual arbitrariness space $Q_m z_m$ can be further used for assigning motion priorities between the arm and the vehicle, for instance, via the following additional least-priority ending task

$$y = \underset{y \in S_m}{\operatorname{argmin}} \| v \|^2 = \rho_{m+1}$$

whose solution ρ_{m+1} (with no more arbitrariness required) finally assures (while respecting all previous priorities) a motion minimality of the vehicle, thus implicitly assigning to the arm a greater mobility, which in turn allows the exploitation of its generally higher motion precision, especially during the ultimate convergence toward the final grasping.

Implementations

The recently realized TRIDENT system of Fig. 1, embedding the above introduced task-priority-based control architecture, has been operating at sea in 2012 (Port Soller Harbor, Mallorca, Spain). A detailed presentation of the preliminary performed simulations, then followed by pool experiments, and finally followed by field trials executed within a true underwater sea environment can be found in Simetti et al. (2013). The related EU-funded TRIDENT project (Sanz et al. 2012) is the first one where agile manipulation could be effectively achieved by part of an underwater floating manipulator, not only as the consequence of the comparable masses and inertia exhibited by the vehicle and arm, but mainly due to the adopted unified task-priority-based control framework. Capabilities for autonomous underwater floating manipulation were however already achieved for the first time in 2009 at the University of Hawaii, within the SAUVIM project (Marani et al. 2009, 2014; Yuh et al. 1998) even if without effective agility (the related system was in fact a 6-t vehicle endowed with a less than 35 kg arm).

Future Directions

The presented task-priority-based KCL structure is invariant with the addition, deletion, and substitution (even on-the-fly) of the various objectives, as well as invariant to changes in their priority ordering, thus constituting an invariant core potentially capable of supporting intervention tasks beyond the sole sample collection ones. On this basis, more complex systems and operational cases, such as, for instance, multi-arm systems and/or even cooperating ones, can be foreseen to be developed along the lines established by the roadmap of Fig. 3 (with case 0 the current development state).

The future availability of agile floating single-arm or multi-arm manipulators, also implementing cooperative interventions in force of a unified control and coordination structure (to this aim purposely extended), might in fact pave the way toward the realization of underwater hard-work robotized places, where different intervention agents might individually or cooperatively perform different object manipulation and transportation activities, also including assembly ones, thus far beyond the here considered case of sample collection. Such scenarios deserve the attention not only of the science community when needing to execute underwater works (excavation, coring, instrument handling, etc.,

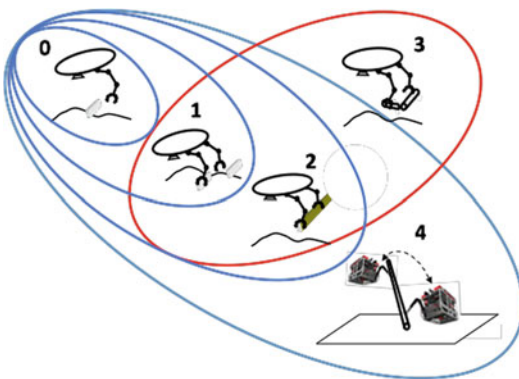other than sample collection) at increasing depths but obviously also those of the offshore industry.

Moreover, by exploiting the current and future developments on underwater exploration and survey mission performed by normal AUVs (i.e., nonmanipulative), a possible work scenario might also include the presence of these lasts, for accomplishing different service activities supporting the intervention ones, for instance, relays with the surface, then informative activities (for instance, the delivery of the area model built during a previous survey phase or the delivery of the intervention mission, both downloaded when in surface and then transferred to the intervention agents upon docking), or even when hovering on the work area (for instance, close to a well-recognized feature) behaving as a local reference system for the self-localization of the operative agents via twin USBL devices.

Cross-References

▶ Control of Networks of Underwater Vehicles
▶ Control of Ship Roll Motion
▶ Dynamic Positioning Control Systems for Ships and Underwater Vehicles
▶ Mathematical Models of Marine Vehicle-Manipulator Systems
▶ Mathematical Models of Ships and Underwater Vehicles
▶ Motion Planning for Marine Control Systems
▶ Redundant Robots
▶ Robot Grasp Control
▶ Robot Teleoperation
▶ Underactuated Marine Control Systems
▶ Underactuated Robots

Advanced Manipulation for Underwater Sampling, Fig. 3 A sketch of the foreseen roadmap for future development of marine intervention robotics

Bibliography

Antonelli G (2006) Underwater robotics. Springer tracts in advanced robotics. Springer, New York

Casalino G (2011) Trident overall system modeling, including all needed variables for reactive coordination. Technical report ISME-2011. Available at http://www.grasal.dist.unige.it/files/89

Casalino G, Zereik E, Simetti E, Torelli S, Sperindè A, Turetta A (2012a) Agility for uunderwater floating manipulation task and subsystem priority based control strategies. In: International conference on intelligent robots and systems (IROS 2012), Vilamoura-Algarve

Casalino G, Zereik E, Simetti E, Torelli S, Sperindè A, Turetta A (2012b) A task and subsystem priority basedCcontrol strategy for underwater floating manipulators. In: IFAC workshop on navigation, guidance and control of underwater vehicles (NGCUV 2012), Porto

Marani G, Choi SK, Yuh J (2009) Underwater autonomous manipulation for intervention missions AUVs. Ocean Eng 36(1):15–23

Marani G, Yuh J (2014) Introduction to autonomous manipulation - case study with an underwater robot, SAUVIM. Springer Tracts in Advanced Robotics 102, Springer, pp. 1-156

Nakamura Y (1991) Advanced robotics: redundancy and optimization. Addison Wesley, Reading

Sanz P, Ridao P, Oliver G, Casalino G, Insurralde C, Silvestre C, Melchiorri M, Turetta A (2012) TRIDENT: recent I mprovements about autonomous underwater intervention missions. In: IFAC workshop on navigation, guidance and control of underwater vehicles (NGCUV 2012), Porto

Simetti E, Casalino G, Torelli S, Sperinde A, Turetta A (2013) Experimental results on task priority and dynamic programming based approach to underwater floating manipulation. In: OCEANS 2013, Bergen, June 2013

Yoshikawa T (1985) Manipulability of robotic mechanisms. Int J Robot Res 4(1):3–9. 1998

Yuh J, Cho SK, Ikehara C, Kim GH, McMurty G, Ghasemi-Nejhad M, Sarkar N, Sugihara K (1998) Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM). In: Proceedings of the 1998 international symposium on underwater technology, Tokyo, Apr 1998

Air Traffic Management Modernization: Promise and Challenges

Christine Haissig
Honeywell International Inc., Minneapolis, MN, USA

Synonyms

ATM Modernization

Abstract

This entry provides a broad overview of how air traffic for commercial air travel is scheduled and managed throughout the world. The major causes of delays and congestion are described, which include tight scheduling, safety restrictions, infrastructure limitations, and major disturbances. The technical and financial challenges to air traffic management are outlined, along with some of the promising developments for future modernization.

Keywords

Air traffic management; Air traffic control; Airport capacity; Airspace management; Flight safety

Introduction: How Does Air Traffic Management Work?

This entry focuses on air traffic management for commercial air travel, the passenger- and cargo-carrying operations with which most of us are familiar. This is the air travel with a pressing need for modernization to address current and future congestion. Passenger and cargo traffic is projected to double over the next 20 years, with growth rates of 3–4 % annually in developed markets such as the USA and Europe and growth rates of 6 % and more in developing markets such as Asia Pacific and the Middle East.

In most of the world, air travel is a distributed, market-driven system. Airlines schedule flights based on when people want to fly and when it is optimal to transport cargo. Most passenger flights are scheduled during the day; most package carrier flights are overnight. Some airports limit how many flights can be scheduled by having a slot system, others do not. This decentralized schedule of flights to and from airports around the world is controlled by a network of air navigation service providers (ANSPs) staffed with air traffic controllers, who ensure that aircraft are separated safely.

The International Civil Aviation Organization (ICAO) has divided the world's airspace into flight information regions. Each region has a country that controls the airspace, and the ANSP for each country can be a government department, state-owned company, or private organization. For example, in the United States, the ANSP is the Federal Aviation Administration (FAA), which is a government department. The Canadian ANSP is NAV CANADA, which is a private company.

Each country is different in terms of the services provided by the ANSP, how the ANSP operates, and the tools available to the controllers. In the USA and Europe, the airspace is divided into sectors and areas around airports. An air traffic control center is responsible for traffic flow within its sector and rules and procedures are in place to cover transfer of control between sectors. The areas around busy airports are usually handled by a terminal radar approach control. The air traffic control tower personnel handle departing aircraft, landing aircraft, and the movement of aircraft on the airport surface.

Air traffic controllers in developed air travel markets like the USA and Europe have tools that help them with the business of controlling and separating aircraft. Tower controllers operating at airports can see aircraft directly through windows or on computer screens through surveillance technology such as radar and Automatic Dependent Surveillance-Broadcast (ADS-B). Tower controllers may have additional tools to help detect and prevent potential collisions on the airport surface. En route controllers can see aircraft on computer screens and may have additional tools to help detect potential losses of separation between aircraft. Controllers can communicate with aircraft via radio and some have datalink communication available such as Controller-Pilot Datalink Communications (CPDLC).

Flight crews have tools to help with navigating and flying the airplane. Autopilots and autothrottles off-load the pilot from having to continuously control the aircraft; instead the pilot can specify the speed, altitude, and heading and the autopilot and autothrottle will maintain those commands.

Flight management systems (FMS) assist in flight planning in addition to providing lateral and vertical control of the airplane. Many aircraft have special safety systems such as the Traffic Alert and Collision Avoidance System, which alerts the flight crew to potential collisions with other airborne aircraft, and the Terrain Avoidance Warning Systems (TAWS), which alert the flight crew to potential flight into terrain.

Causes of Congestion and Delays

Congestion and delays are caused by multiple reasons. These include tight scheduling, safety limitations on how quickly aircraft can take off and land and how closely they can fly, infrastructure limitations such as the number of runways at an airport and the airway structure, and disturbances such as weather and unscheduled maintenance.

Tight Scheduling

Tight scheduling is a major contributor to congestion and delays. The hub and spoke system that many major airlines operate with to minimize connection times means that aircraft arrive and depart in multiple banks during the day. During the arrival and departure banks, airports are very busy. As mentioned previously, passengers have preferred times to travel, which also increase demand at certain times. At airports that do not limit flight schedules by using slot scheduling, the number of flights scheduled can actually exceed the departure and arrival capacity of the airport even in best-case conditions. One of the reasons that airlines are asked to report on-time statistics is to make the published airline schedules more reflective of the average time from departure to arrival, not the best-case time.

Aircraft themselves are also tightly scheduled. Aircraft are an expensive capital asset. Since customers are very sensitive to ticket prices, airlines need to have their aircraft flying as many hours as possible per day. Airlines also limit the number of spare aircraft and flight crews available to fill in when operations are disrupted to control costs.

Safety Restrictions

Safety restrictions contribute to congestion. There is a limit to how quickly aircraft can take off from and land on a runway. Sometimes runways are used for both departing and arriving aircraft; at other times a runway may be used for departures only or arrivals only. Either way, the rule that controllers follow for safety is that only one aircraft can occupy the runway at one time. Thus, a landing aircraft must turn off of the runway before another aircraft can take off or land. This limitation and other limitations like the ability of controllers to manage the arrival and departure aircraft propagate backwards from the airport. Aircraft need to be spaced in an orderly flow and separated no closer than what can be supported by airport arrival rates. The backward propagation can go all the way to the departure airports and cause aircraft to be held on the ground as a means to regulate the traffic flow into a congested airport or through a congested air traffic sector.

There is a limit on how close aircraft can fly. Aircraft produce a wake that can be dangerous for other aircraft that are following too closely behind. Pilots are aware of this limitation and space safely when doing visual separation. Rules that controllers apply for separation take into account wake turbulence limitations, surveillance limitations, and limitations on how well aircraft can navigate and conform to the required speed, altitude, and heading.

The human is a safety limitation. Controllers and pilots are human. Being human, they have excellent reasoning capability. However, they are limited as to the number of tasks they can perform and are subject to fatigue. The rules and procedures in place to manage and fly aircraft take into account human limitations.

Infrastructure Limitations

Infrastructure limitations contribute to congestion and delays. Airport capacity is one infrastructure limitation. The number of runways combined with the available aircraft gates and capacity to process passengers through the terminal limit the airport capacity.

The airspace itself is a limitation. The airspace where controllers provide separation services is divided into an orderly structure of airways. The airways are like one-way, one-lane roads in the sky. They are stacked at different altitudes, which are usually separated by either 1,000 ft. or 2,000 ft. The width of the airways depends on how well aircraft can navigate. In the US domestic airspace where there are regular navigation aids and direct surveillance of aircraft, the airways have a plus or minus 4 NM width. Over the ocean, airways may need to be separated laterally by as much as 120 NM since there are fewer navigation aids and aircraft are not under direct control but separated procedurally. The limited number of airways that the airspace can support limits available capacity.

The airways themselves have capacity limitations just as traditional roads do. There are special challenges for airways since aircraft need a minimum separation distance, aircraft cannot slow down to a stop, and airways do not allow passing. So, although it may look like there is a lot of space in which aircraft can fly, there are actually a limited number of routes between a city pair or over oceanic airspace.

The radio that is used for pilots and controllers to communicate is another infrastructure limitation. At busy airports, there is significant radio congestion and pilots may need to wait to get an instruction or response from a controller.

Disturbances

Weather is a significant disturbance in air traffic management. Weather acts negatively in many ways. Wet or icy pavement affects the braking ability of aircraft so they cannot vacate a runway as quickly as in dry conditions. Low cloud ceilings mean that all approaches must be instrument approaches rather than visual approaches, which also reduces runway arrival rates. Snow must be cleared from runways, closing them for some period of time. High winds can mean that certain approaches cannot be used because they are not safe. In extreme weather, an airport may need to close. Weather can block certain airways from use, requiring rerouting of aircraft. Rerouting increases demand on nearby airways, which may

or may not have the required additional capacity, so the rerouting cascades on both sides of the weather.

Why Is Air Traffic Management Modernization So Hard?

Air traffic management modernization is difficult for financial and technical reasons. The air traffic management system operates around the clock. It cannot be taken down for a significant period of time without a major effect on commerce and the economy.

Financing is a significant challenge for air traffic management modernization. Governments worldwide are facing budgetary challenges and improvements to air travel are one of many competing financial interests. Local airport authorities have similar challenges in raising money for airport improvements. Airlines have competitive limitations on how much ticket prices can rise and therefore need to see a payback on investment in aircraft upgrades that can be as short as 2 years.

Another financial challenge is that the entity that needs to pay for the majority of an improvement may not be the entity that gets the majority of the benefit, at least near term. One example of this is the installation of ADS-B transmitters on aircraft. Buying and installing an ADS-B transmitter costs the aircraft owner money. It benefits the ANSPs, who can receive the transmissions and have them augment or replace expensive radar surveillance, but only if a large number of aircraft are equipped. Eventually the ANSP benefit will be seen by the aircraft operator through lower operating costs but it takes time. This is one reason that ADS-B transmitter equipage was mandated in the USA, Europe, and other parts of the world rather than letting market forces drive equipage.

All entities, whether governmental or private, need some sort of business case to justify investment, where it can be shown that the benefit of the improvement outweighs the cost. The same system complexity that makes congestion and delays in one region propagate throughout the system

makes it a challenge to accurately estimate benefits. It is complicated to understand if an improvement in one part of the system will really help or just shift where the congestion points are. Decisions need to be made on what improvements are the best to invest in. For government entities, societal benefits can be as important as financial payback, and someone needs to decide whose interests are more important. For example, the people living around an airport might want longer arrival paths at night to minimize noise while air travelers and the airline want the airline to fly the most direct route into an airport. A combination of subject matter expertise and simulation can provide a starting point to estimate benefit, but often only operational deployment will provide realistic estimates.

It is a long process to develop new technologies and operational procedures even when the benefit is clear and financing is available. The typical development steps include describing the operational concept; developing new controllers procedures, pilot procedures, or phraseology if needed; performing a safety and performance analysis to determine high level requirements; performing simulations that at some point may include controllers or pilots; designing and building equipment that can include software, hardware, or both; and field testing or flight testing the new equipment. Typically, new ground tools are field tested in a shadow mode, where controllers can use the tool in a mock situation driven by real data before the tool is made fully operational. Flight testing is performed on aircraft that are flying with experimental certificates so that equipment can be tested and demonstrated prior to formal certification.

Avionics need to be certified before operational use to meet the rules established to ensure that a high safety standard is applied to air travel. To support certification, standards are developed. Frequently the standards are developed through international cooperation and through consensus decision-making that includes many different organizations such as ANSPs, airlines, aircraft manufacturers, avionics suppliers, pilot associations, controller associations, and more. This is a slow process but an important one, since it

reduces development risk for avionics suppliers and helps ensure that equipment can be used worldwide.

Once new avionics or ground tools are available, it takes time for them to be deployed. For example, aircraft fleets are upgraded as aircraft come in for major maintenance rather than pulling them out of scheduled service. Flight crews need to be trained on new equipment before it can be used, and training takes time. Ground tools are typically deployed site by site, and the controllers also require training on new equipment and new procedures.

Promise for the Future

Despite the challenges and complexity of air traffic management, there is a path forward for significant improvement in both developed and developing air travel markets. Developing air travel markets in countries like China and India can improve air traffic management using procedures, tools, and technology that is already used in developed markets such as the USA and Europe. Emerging markets like China are willing to make significant investments in improving air traffic management by building new airports, expanding existing airports, changing controller procedures, and investing in controller tools. In developed markets, new procedures, tools, and technologies will need to be implemented. In some regions, mandates and financial incentives may play a part in enabling infrastructure and equipment changes that are not driven by the marketplace.

The USA and Europe are both supporting significant research, development, and implementation programs to support air traffic management modernization. In the USA, the FAA has a program known as NextGen, the Next Generation Air Transportation System. In Europe, the European Commission oversees a program known as SESAR, the Single European Sky Air Traffic Management Research, which is a joint effort between the European Union, EUROCONTROL, and industry partners. Both programs have substantial support and

financing. Each program has organized its efforts differently but there are many similarities in the operational objectives and improvements being developed.

Airport capacity problems are being addressed in multiple ways. Controllers are being provided with advanced surface movement guidance and control systems that combine radar surveillance, ADS-B surveillance, and sensors installed at the airport with valued-added tools to assist with traffic control and alert controllers to potential collisions. Datalink communications between controllers and pilots will reduce radio-frequency congestion, reduce communication errors, and enable more complex communication. The USA and Europe have plans to develop a modernized datalink communication infrastructure between controllers and pilots that would include information like departure clearances and the taxiway route clearance. Aircraft on arrival to an airport will be controlled more precisely by equipping aircraft with capabilities such as the ability to fly to a required time of arrival and the ability to space with respect to another aircraft.

Domestic airspace congestion is being addressed in Europe by moving towards a single European sky where the ANSPs for the individual nations coordinate activities and airspace is structured not as 27 national regions but operated as larger blocks. Similar efforts are undergoing in the USA to improve the cooperation and coordination between the individual airspace sectors. In some countries, large blocks of airspace are reserved for special use by the military. In those countries, efforts are in place to have dynamic special use airspace that is reserved on an as-needed basis but otherwise available for civil use.

Oceanic airspace congestion is being addressed by leveraging the improved navigation performance of aircraft. Some route structures are available only to aircraft that can flight to a required navigation performance. These route structures have less required lateral separation, and thus more routes can be flown in the same airspace. Pilot tools that leverage ADS-B are allowing aircraft to make flight level changes with reduced separation and in the future

are expected to allow pilots to do additional maneuvering that is restricted today, such as passing slower aircraft.

Weather cannot be controlled but efforts are underway to do better prediction and provide more accurate and timely information to pilots, controllers, and aircraft dispatchers at airlines. On-board radars that pilots use to divert around weather are adding more sophisticated processing algorithms to better differentiate hazardous weather. Future flight management systems will have the capability to include additional weather information. Datalinks between the air and the ground or between aircraft may be updated to include information from the on-board radar systems, allowing aircraft to act as local weather sensors. Improved weather information for pilots, controllers, and dispatchers improves flight planning and minimizes the necessary size of deviations around hazardous weather while retaining safety.

Weather is also addressed by providing aircraft and airports with equipment to improve airport access in reduced visibility. Ground-based augmentation systems installed at airports provide aircraft with the capability to do precision-based navigation for approaches to airports with low weather ceilings. Other technologies like enhanced vision and synthetic vision, which can be part of a combined vision system, provide the capability to land in poor visibility.

Summary

Air traffic management is a complex and interesting problem. The expected increase in air travel worldwide is driving a need for improvements to the existing system so that more passengers can be handled while at the same time reducing congestion and delays. Significant research and development efforts are underway worldwide to develop safe and effective solutions that include controller tools, pilot tools, aircraft avionics, infrastructure improvements, and new procedures. Despite the technical and financial challenges, many promising technologies and new procedures will be implemented in the near, mid-, and far term to support air traffic management modernization worldwide.

Cross-References

▶ Aircraft Flight Control
▶ Pilot-Vehicle System Modeling

Bibliography

Collinson R (2011) Introduction to avionics systems, 3rd edn. Springer, Dordrecht
http://www.faa.gov/nextgen/
http://www.sesarju.eu/
Nolan M (2011) Fundamentals of air traffic control, 5th edn. Cengage Learning, Clifton Park

Aircraft Flight Control

Dale Enns
Honeywell International Inc., Minneapolis, MN, USA

Abstract

Aircraft flight control is concerned with using the control surfaces to change aerodynamic moments, to change attitude angles of the aircraft relative to the air flow, and ultimately change the aerodynamic forces to allow the aircraft to achieve the desired maneuver or steady condition. Control laws create the commanded control surface positions based on pilot and sensor inputs. Traditional control laws employ proportional and integral compensation with scheduled gains, limiting elements, and cross feeds between coupled feedback loops. Dynamic inversion is an approach to develop control laws that systematically addresses the equivalent of gain schedules and the multivariable cross feeds, can incorporate constrained optimization for the limiting elements, and maintains the use of proportional and integral compensation to achieve the benefits of feedback.

Keywords

Control allocation; Control surfaces; Dynamic inversion; Proportional and integral control; Rigid body equations of motion; Zero dynamics

Introduction

Flying is made possible by flight control and this applies to birds and the Wright Flyer, as well as modern flight vehicles. In addition to balancing lift and weight forces, successful flight also requires a balance of moments or torques about the mass center. Control is a means to adjust these moments to stay in equilibrium and to perform maneuvers. While birds use their feathers and the Wright Flyer warped its wings, modern flight vehicles utilize hinged control surfaces to adjust the moments. The control action can be open or closed loop, where closed loop refers to a feedback loop consisting of sensors, computer, and actuation. A direct connection between the cockpit pilot controls and the control surfaces without a feedback loop is open loop control. The computer in the feedback loop implements a control law (computer program). The development of the control law is discussed in this entry.

Although the following discussion is applicable to a wide range of flight vehicles including gliders, unmanned aerial vehicles, lifting bodies, missiles, rockets, helicopters, and satellites, the focus of this entry will be on fixed wing commercial and military aircraft with human pilots.

Flight

Aircraft are maneuvered by changing the forces acting on the mass center, e.g., a steady level turn requires a steady force towards the direction of turn. The force is the aerodynamic lift force (L) and it is banked or rotated into the direction of the turn. The direction can be adjusted with the bank angle (μ) and for a given airspeed (V) and air density (ρ), the magnitude of the force can be adjusted with the angle-of-attack (α). This is called bank-to-turn. Aircraft, e.g., missiles, can also skid-to-turn where the aerodynamic side force (Y) is adjusted with the sideslip angle (β) but this entry will focus on bank-to-turn.

Equations of motion (Enns et al. 1996; Stevens and Lewis 1992) can be used to relate the time rates of change of μ, α, and β to roll (p), pitch (q), and yaw (r) rate. See Fig. 1. Approximate relations (for near steady level flight with no wind) are

Aircraft Flight Control, Fig. 1 Flight control variables

$$\dot{\mu} = \mathsf{p}$$

$$\dot{\alpha} = q + \frac{L - m\mathsf{g}}{mV}$$

$$\dot{\beta} = -r + \frac{Y}{mV}$$

where m is the aircraft mass, and g is the gravitational acceleration. In straight and level flight conditions $L = mg$ and $Y = 0$ so we think of these equations as kinematic equations where the rates of change of the angles μ, α, and β are the angular velocities p, q, and r.

Three moments called roll, pitch, and yaw for angular motion to move the right wing up or down, nose up or down, and nose right or left, respectively create the angular accelerations to change p, q, and r, respectively. The equations are Newton's 2nd law for rotational motion. The moments (about the mass center) are dominated by aerodynamic contributions and depend on ρ, V, α, β, p, q, r, and the control surfaces. The control surfaces are aileron (δ_a), elevator (δ_e), and rudder (δ_r) and are arranged to contribute primarily roll, pitch, and yaw moments respectively.

The control surfaces (δ_a, δ_e, δ_r) contribute to angular accelerations which are integrated to obtain the angular rates (p, q, r). The integral of angular rates contributes to the attitude angles (μ, α, β). The direction and magnitude of aerodynamic forces can be adjusted with the attitude angles. The forces create the maneuvers or steady conditions for operation of the aircraft.

Pure Roll Axis Example
Consider just the roll motion. The differential equation (Newton's 2nd law for the roll degree-of-freedom) for this dynamical system is

$$\dot{\mathsf{p}} = L_{\mathsf{p}}\mathsf{p} + L_{\delta_a}\delta_a$$

where L_p is the stability derivative and $L_{\delta a}$ is the control derivative both of which can be regarded as constants for a given airspeed and air density.

Pitch Axis or Short Period Example
Consider just the pitch and heave motion. The differential equations (Newton's 2nd law for the

pitch and heave degrees-of-freedom) for this dynamical system are

$$\dot{q} = M_\alpha \alpha + M_q q + M_{\delta_e}\delta_e$$

$$\dot{\alpha} = Z_\alpha \alpha + q + Z_{\delta_e}\delta_e$$

where M_α, M_q, Z_α are stability derivatives, and $M_{\delta e}$ is the control derivative, all of which can be regarded as constants for a given airspeed and air density.

Although $Z_\alpha < 0$ and $M_q < 0$ are stabilizing, $M_\alpha > 0$ makes the short period motion inherently unstable. In fact, the short period motion of the Wright Flyer was unstable. Some modern aircraft are also unstable.

Lateral-Directional Axes Example
Consider just the roll, yaw, and side motion with four state variables (μ, p, r, β) and two inputs (δ_a, δ_r). We will use the standard state space equations with matrices A, B, C for this example.

The short period equations apply for yaw and side motion (or dutch roll motion) with appropriate replacements, e.g., q with r, α with $-\beta$, M with N. We add the term $V^{-1}g\mu$ to the $\dot{\beta}$ equation. We include the kinematic equation $\dot{\mu} = p$ and add the term $L_\beta\beta$ to the $\dot{p}$ equation. The dutch roll, like the short period, can be unstable if $N_\beta < 0$, e.g., airplanes without a vertical tail.

There is coupling between the motions associated with stability derivatives L_r, L_β, N_p and control derivatives $L_{\delta r}$ and $N_{\delta a}$. This is a fourth order multivariable coupled system where δ_a, δ_r are the inputs and we can consider (p, r) or (μ, β) as the outputs.

Control

The control objectives are to provide stability, disturbance rejection, desensitization, and satisfactory steady state and transient response to commands. Specifications and guidelines for these objectives are assessed quantitatively with frequency, time, and covariance analyses and simulations.
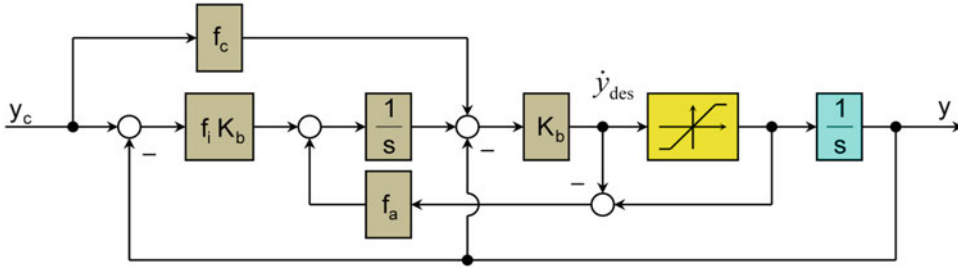
Aircraft Flight Control, Fig. 2 Closed loop feedback system and desired dynamics

Integrator with P + I Control

The system to be controlled is the integrator for y in Fig. 2 and the output of the integrator (y) is the controlled variable. The proportional gain ($K_b > 0$) is a frequency and sets the bandwidth or crossover frequency of the feedback loop. The value of K_b will be between 1 and 10 rad/s in most aircraft applications. Integral action can be included with the gain, $f_i > 0$ with a value between 0 and 1.5 in most applications. The value of the command gain, $f_c > 0$, is set to achieve a desired closed loop response from the command y_c to the output y. Values of $f_i = 0.25$ and $f_c = 0.5$ are typical. In realistic applications, there is a limit that applies at the input to the integrator. In these cases, we are obligated to include an anti-integral windup gain, $f_a > 0$ (typical value of 2) to prevent continued integration beyond the limit. The input to the limiter ($\dot{y}_{des}$) is called the desired rate of change of the controlled variable (Enns et al. 1996).

The closed loop transfer function is

$$\frac{y}{y_c} = \frac{K_b(f_c s + f_i K_b)}{s^2 + K_b s + f_i K_b^2}$$

and the pilot produces the commands, (y_c) with cockpit inceptors, e.g., sticks, pedals.

The control system robustness can be adjusted with the choices made for y, K_b, f_i, and f_c.

These desired dynamics are utilized in all of the examples to follow. In the following, we use dynamic inversion (Enns et al. 1996; Wacker et al. 2001) to algebraically manipulate the equations of motion into the equivalent of the integrator for y in Fig. 2.

Pure Roll Motion Example

With algebraic manipulations called dynamic inversion we can use the pure integrator results in the previous section for the pure roll motion example. For the controlled variable $y = p$, given a measurement of the state $x = p$ and values for L_p and $L_{\delta a}$, we simply solve for the input ($u = \delta_a$) that gives the desired rate of change of the output $\dot{y}_{des} = \dot{p}_{des}$. The solution is

$$\delta_a = L_{\delta_a}^{-1}(\dot{p}_{des} - L_p p)$$

Since $L_{\delta a}$ and L_p vary with air density and airspeed, we are motivated to schedule these portions of the control law accordingly.

Short Period Example

Similar algebraic manipulations use the general state space notation

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

We want to solve for u to achieve a desired rate of change of y, so we start with

$$\dot{y} = CAx + CBu$$

If we can invert CB, i.e., it is not zero, for the short period case, we solve for u with

$$u = (CB)^{-1}(\dot{y}_{des} - CAx)$$

Implementation requires a measurement of the state, x and models for the matrices CA and CB.

The closed loop poles include the open loop zeros of the transfer function $\frac{y(s)}{u(s)}$ (zero dynamics) in addition to the roots of the desired dynamics characteristic equation. Closed loop stability requires stable zero dynamics. The zero dynamics have an impact on control system robustness and can influence the precise choice of y.

When $y = q$, the control law includes the following dynamic inversion equation

$$\delta_e = M_{\delta_e}^{-1}(\dot{q}_{\text{des}} - M_q q - M_\alpha \alpha)$$

and the open loop zero is $Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha$, which in almost every case of interest is a negative number.

Note that there are no restrictions on the open loop poles. This control law is effective and practical in stabilization of an aircraft with an open loop unstable short period mode.

Since $M_{\delta e}$, M_q and M_α vary with air density and airspeed we are motivated to schedule these portions of the control law accordingly.

When $y = \alpha$, the zero dynamics are not suitable as closed loop poles. In this case, the pitch rate controller described above is the inner loop and we apply dynamic inversion a second time as an outer loop (Enns and Keviczky 2006) where we approximate the angle-of-attack dynamics with the simplification that pitch rate has reached steady state, i.e., $\dot{q} = 0$ and regard pitch rate as the input ($u = q$) and angle-of-attack as the controlled variable ($y = \alpha$). The approximate equation of motion is

$$\dot{\alpha} = Z_\alpha \alpha + q - Z_{\delta_e} M_{\delta_e}^{-1} \left(M_\alpha \alpha + M_q q \right)$$
$$= \left(Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha \right) \alpha$$
$$+ \left(1 - Z_{\delta_e} M_{\delta_e}^{-1} M_q \right) q$$

This equation is inverted to give

$$q_c = \left(1 - Z_{\delta_e} M_{\delta_e}^{-1} M_q \right)^{-1}$$
$$\left[\dot{\alpha}_{\text{des}} - \left(Z_\alpha - Z_{\delta_e} M_{\delta_e}^{-1} M_\alpha \right) \alpha \right]$$

q_c obtained from this equation is passed to the inner loop as a command, i.e., y_c of the inner loop.

Lateral-Directional Example

If we choose the two angular rates as the controlled variables (p, r), then the zero dynamics are favorable. We use the same proportional plus integral desired dynamics in Fig. 2 but there are two signals represented by each wire (one associated with p and the other r).

The same state space equations are used for the dynamic inversion step but now CA and CB are 2×4 and 2×2 matrices, respectively instead of scalars. The superscript in $u = (CB)^{-1}(\dot{y}_{\text{des}} - CAx)$ now means matrix inverse instead of reciprocal. The zero dynamics are assessed with the transmission zeros of the matrix transfer function $(p, r)/(\delta_a, \delta_r)$.

In the practical case where the aileron and rudder are limited, it is possible to place a higher priority on solving one equation vs. another if the equations are coupled, by proper allocation of the commands to the control surfaces which is called control allocation (Enns 1998). In these cases, we use a constrained optimization approach

$$\min_{u_{\text{min}} \leq u \leq u_{\text{max}}} ||CBu - (\dot{y}_{\text{des}} - CAx)||$$

instead of the matrix inverse followed by a limiter. In cases where there are redundant controls, i.e., the matrix CB has more columns than rows, we introduce a preferred solution, u_p and solve a different constrained optimization problem

$$\min_{CBu+CAx=\dot{y}_{\text{des}}} ||u - u_p||$$

to find the solution that solves the equations that is closest to the preferred solution. We utilize weighted norms to accomplish the desired priority.

An outer loop to control the attitude angles (μ, β) can be obtained with an approach analogous to the one used in the previous section.

Nonlinear Example

Dynamic inversion can be used directly with the nonlinear equations of motion (Enns et al. 1996; Wacker et al. 2001). General equations of motion, e.g., 6 degree-of-freedom rigid body can be expressed with $\dot{x} = f(x, u)$ and the controlled

variable is given by $y = h(x)$. With the chain rule of calculus we obtain

$$\dot{y} = \frac{\partial h}{\partial x}(x)\, f(x, u)$$

and for a given $\dot{y} = \dot{y}_{\text{des}}$ and (measured) x we can solve this equation for u either directly or approximately. In practice, the first order Taylor Series approximation is effective

$$\dot{y} \cong a(x, u_0) + b(x, u_0)(u - u_0)$$

where u_0 is typically the past value of u, in a discrete implementation. As in the previous example, Fig. 2 can be used to obtain $\dot{y}_{\text{des}}$. The terms $a(x, u_0) - b(x, u_0)\, u_0$ and $b(x, u_0)$ are analogous to the terms CAx and the matrix CB, respectively. Control allocation can be utilized in the same way as discussed above. The zero dynamics are evaluated with transmission zeros at the intended operating points. Outer loops can be employed in the same manner as discussed in the previous section.

The control law with this approach utilizes the equations of motion which can include table lookup for aerodynamics, propulsion, mass properties, and reference geometry as appropriate. The raw aircraft data or an approximation to the data takes the place of gain schedules with this approach.

portions of the control law are scheduled. Aircraft do exhibit coupling between axes and so multivariable feedback loop approaches are effective. Nonlinearities in the form of limits (noninvertible) and nonlinear expressions, e.g., trigonometric, polynomial, and table look-up (invertible) are present in flight control development. The dynamic inversion approach has been shown to include the traditional feedback control principles, systematically develops the equivalent of the gain schedules, applies to multivariable systems, applies to invertible nonlinearities, and can be used to avoid issues with noninvertible nonlinearities to the extent it is physically possible.

Future developments will include adaptation, reconfiguration, estimation, and nonlinear analyses. Adaptive control concepts will continue to mature and become integrated with approaches such as dynamic inversion to deal with unstructured or nonparameterized uncertainty or variations in the aircraft dynamics. Parameterized uncertainty will be incorporated with near real time reconfiguration of the aircraft model used as part of the control law, e.g., reallocation of control surfaces after an actuation failure. State variables used as measurements in the control law will be estimated as well as directly measured in nominal and sensor failure cases. Advances in nonlinear dynamical systems analyses will create improved intuition, understanding, and guidelines for control law development.

Summary and Future Directions

Flight control is concerned with tracking commands for angular rates. The commands may come directly from the pilot or indirectly from the pilot through an outer loop, where the pilot directly commands the outer loop. Feedback control enables stabilization of aircraft that are inherently unstable and provides disturbance rejection and insensitive closed-loop response in the face of uncertain or varying vehicle dynamics. Proportional and integral control provide these benefits of feedback. The aircraft dynamics are significantly different for low altitude and high speed compared to high altitude and low speed and so

Cross-References

▶ PID Control
▶ Satellite Control
▶ Tactical Missile Autopilots

Bibliography

Enns DF (1998) Control allocation approaches. In: Proceedings of the AIAA guidance, navigation, and control conference, Boston

Enns DF, Keviczky T (2006) Dynamic inversion based flight control for autonomous RMAX helicopter. In: Proceedings of the 2006 American control conference, Minneapolis, 14–16 June 2006

Enns DF et al (1996) Application of multivariable control theory to aircraft flight control laws, final report: multivariable control design guidelines. Technical report WL-TR-96-3099, Flight Dynamics Directorate, Wright-Patterson Air Force Base, OH 45433-7562, USA

Stevens BL, Lewis FL (1992) Aircraft control and simulation. Wiley, New York

Wacker R, Enns DF, Bugajski DJ, Munday S, Merkle S (2001) X-38 application of dynamic inversion flight control. In: Proceedings of the 24th annual AAS guidance and control conference, Breckenridge, 31 Jan–4 Feb 2001

Applications of Discrete-Event Systems

Spyros Reveliotis
School of Industrial & Systems Engineering,
Georgia Institute of Technology, Atlanta,
GA, USA

Abstract

This entry provides an overview of the problems addressed by discrete-event systems (DES) theory, with an emphasis on their connection to various application contexts. The primary intentions are to reveal the caliber and the strengths of this theory and to direct the interested reader, through the listed citations, to the corresponding literature. The concluding part of the entry also identifies some remaining challenges and further opportunities for the area.

Keywords

Applications; Discrete-event systems

Introduction

Discrete-event systems (DES) theory (▶ Models for Discrete Event Systems: An Overview) (Cassandras and Lafortune 2008) emerged in the late 1970s/early 1980s from the effort of the controls community to address the control needs of applications concerning some complex production and service operations, like those taking place in manufacturing and other workflow systems, telecommunication and data-processing systems, and transportation systems. These operations were seeking the ability to support higher levels of efficiency and productivity and more demanding notions of quality of product and service. At the same time, the thriving computing technologies of the era, and in particular the emergence of the microprocessor, were cultivating, and to a significant extent supporting, visions of ever-increasing automation and autonomy for the aforementioned operations. The DES community set out to provide a systematic and rigorous understanding of the dynamics that drive the aforementioned operations and their complexity, and to develop a control paradigm that would define and enforce the target behaviors for those environments in an effective and robust manner.

In order to address the aforementioned objectives, the controls community had to extend its methodological base, borrowing concepts, models, and tools from other disciplines. Among these disciplines, the following two played a particularly central role in the development of the DES theory: (i) the Theoretical Computer Science (TCS) and (ii) the Operations Research (OR). As a new research area, DES thrived on the analytical strength and the synergies that resulted from the rigorous integration of the modeling frameworks that were borrowed from TCS and OR. Furthermore, the DES community substantially extended those borrowed frameworks, bringing in them many of its control-theoretic perspectives and concepts.

In general, DES-based approaches are characterized by (i) their emphasis on a rigorous and formal representation of the investigated systems and the underlying dynamics; (ii) a double focus on time-related aspects and metrics that define traditional/standard notions of performance for the considered systems, but also on a more behaviorally oriented analysis that is necessary for ensuring fundamental notions of "correctness," "stability," and "safety" of the system operation,

especially in the context of the aspired levels of autonomy; (iii) the interplay between the two lines of analysis mentioned in item (ii) above and the further connection of this analysis to structural attributes of the underlying system; and (iv) an effort to complement the analytical characterizations and developments with design procedures and tools that will provide solutions provably consistent with the posed specifications and effectively implementable within the time and other resource constraints imposed by the "real-time" nature of the target applications.

The rest of this entry overviews the current achievements of DES theory with respect to (w.r.t.) the different classes of problems that have been addressed by it and highlights the potential that is defined by these achievements for a range of motivating applications. On the other hand, the constricted nature of this entry does not allow an expansive treatment of the aforementioned themes. Hence, the provided coverage is further supported and supplemented by an extensive list of references that will connect the interested reader to the relevant literature.

A Tour of DES Problems and Applications

DES-Based Behavioral Modeling, Analysis, and Control

The basic characterization of behavior in the DES-theoretic framework is through the various event sequences that can be generated by the underlying system. Collectively, these sequences are known as the (formal) language generated by the plant system, and the primary intention is to restrict the plant behavior within a subset of the generated event strings. The investigation of this problem is further facilitated by the introduction of certain mechanisms that act as formal representations of the studied systems, in the sense that they generate the same strings of events (i.e., the same formal language). Since these models are concerned with the representation of the event sequences that are generated by DES, and not by the exact timing of these events, they are

frequently characterized as *untimed* DES models. In the practical applications of DES theory, the most popular such models are the Finite State Automaton (FSA) (Cassandras and Lafortune 2008; Hopcroft and Ullman 1979; ▶ Supervisory Control of Discrete-Event Systems; ▶ Diagnosis of Discrete Event Systems), and the Petri net (PN) (Cassandras and Lafortune 2008; Murata 1989; ▶ Modeling, Analysis, and Control with Petri Nets).

In the context of DES applications, these modeling frameworks have been used to provide succinct characterizations of the underlying event-driven dynamics and to design controllers, in the form of supervisors, that will restrict these dynamics so that they abide to safety, consistency, fairness, and other similar considerations (▶ Supervisory Control of Discrete-Event Systems). As a more concrete example, in the context of contemporary manufacturing, DES-based behavioral control – frequently referred to as supervisory control (SC) – has been promoted as a systematic methodology for the synthesis and verification of the control logic that is necessary for the support of the, so-called, SCADA (Supervisory Control and Data Acquisition) function. This control function is typically implemented through the Programmable Logic Controllers (PLCs) that have been employed in contemporary manufacturing shop-floors, and DES SC theory can support it (i) by providing more rigor and specificity to the models that are employed for the underlying plant behavior and the imposed specifications and (ii) by offering the ability to synthesize control policies that are provably correct by construction. Some example works that have pursued the application of DES SC along these lines can be found in Balemi et al. (1993), Brandin (1996), Park et al. (1999), Chandra et al. (2003), Endsley et al. (2006), and Andersson et al. (2010).

On the other hand, the aforementioned activity has also defined a further need for pertinent interfaces that will translate (a) the plant structure and the target behavior to the necessary DES-theoretic models and (b) the obtained policies to PLC executables. This need has led to a line of research, in terms of representational models and

computational tools, that is complementary to the core DES developments described in the previous paragraphs. Indicatively we mention the development of GRAFCET (David and Alla 1992) and of the sequential function charts (SFCs) (Lewis 1998) from the earlier times, while some more recent endeavor along these lines is reported in Wightkin et al. (2011) and Alenljung et al. (2012) and the references cited therein.

Besides its employment in the manufacturing domain, DES SC theory has also been considered for the coordination of the communicating processes that take place in various embedded systems (Feng et al. 2007); the systematic validation of the embedded software that is employed in various control applications, ranging from power systems and nuclear plants to aircraft and automotive electronics (Li and Kumar 2012); the synthesis of the control logic in the electronic switches that are utilized in telecom and data networks; and the modeling, analysis, and control of the operations that take place in health-care systems (Sampath et al. 2008). Wassyng et al. (2011) gives a very interesting account of the gains, but also the extensive challenges, experienced by a team of researchers who have tried to apply formal methods, similar to those that have been promoted by the behavioral DES theory, to the development and certification of the software that manages some safety-critical operations for Canadian nuclear plants.

Apart from control, untimed DES models have also been employed for the diagnosis of critical events, like certain failures, that cannot be observed explicitly, but their occurrence can be inferred from some resultant behavioral patterns (Sampath et al. 1996; ▶ Diagnosis of Discrete Event Systems). More recently, the relevant methodology has been extended with prognostic capability (Kumar and Takai 2010), while an interesting variation of it addresses the "dual" problem that concerns the design of systems where certain events or behavioral patterns must remain undetectable by an external observer who has only partial observation of the system behavior; this last requirement has been formally characterized by the notion of "opacity" in the relevant literature, and it finds application

in the design and operation of secure systems (Dubreil et al. 2010; Saboori and Hadjicostis 2012, 2014).

Dealing with the Underlying Computational Complexity

As revealed from the discussion of the previous paragraphs, many of the applications of DES SC theory concern the integration and coordination of behavior that is generated by a number of interacting components. In these cases, the formal models that are necessary for the description of the underlying plant behavior may grow their size very fast, and the algorithms that are involved in the behavioral analysis and control synthesis may become practically intractable. Nevertheless, the rigorous methodological base that underlies DES theory provides also a framework for addressing these computational challenges in an effective and structured manner.

More specifically, DES SC theory provides conditions under which the control specifications can be decomposable to the constituent plant components while maintaining the integrity and correctness of the overall plant behavior (▶ Supervisory Control of Discrete-Event Systems; Wonham 2006). The aforementioned works of Brandin (1996) and Endsley et al. (2006) provide some concrete examples for the application of modular control synthesis. On the other hand, there are fundamental problems addressed by SC theory and practice that require a holistic view of the underlying plant and its operation, and thus, they are not amenable to modular solutions. For such cases, DES SC theory can still provide effective solutions through (i) the identification of special plant structure, of practical relevance, for which the target supervisors are implementable in a computationally efficient manner and (ii) the development of structured approaches that can systematically trade-off the original specifications for computational tractability.

A particular application that has benefited from, and, at the same time, has significantly promoted this last capability of DES SC theory, is that concerning the deadlock-free operation of many systems where a set of processes that execute concurrently and in a staged manner are

competing, at each of their processing stages, for the allocation of a finite set of reusable resources. In DES theory, this problem is known as the liveness-enforcing supervision of sequential resource allocation systems (RAS) (Reveliotis 2005), and it underlies the operation of many contemporary applications: from the resource allocation taking place in contemporary manufacturing shop floors, Ezpeleta et al. (1995), Reveliotis and Ferreira (1996), and Jeng et al. (2002), to the traveling and/or work-space negotiation in robotic systems (Reveliotis and Roszkowska 2011), automated railway (Giua et al. 2006), and other guidepath-based traffic systems (Reveliotis 2000); to Internet-based workflow management systems like those envisioned for e-commerce and certain banking and insurance claim processing applications (Van der Aalst 1997); and to the allocation of the semaphores that control the accessibility of shared resources by concurrently executing threads in parallel computer programs (Liao et al. 2013). A systematic introduction to the DES-based modeling of RAS and their liveness-enforcing supervision is provided in Reveliotis (2005) and Zhou and Fanti (2004), while some more recent developments in the area are epitomized in Reveliotis (2007), Li et al. (2008) and Reveliotis and Nazeem (2013).

Closing the above discussion on the ability of DES theory to address effectively the complexity that underlies the DES SC problem, we should point out that the same merits of the theory have also enabled the effective management of the complexity that underlies problems related to the performance modeling and control of the various DES applications. We shall return to this capability in the next section that discusses the achievements of DES theory in this domain.

DES Performance Control and the Interplay Among Structure, Behavior, and Performance

DES theory is also interested in the performance modeling, analysis, and control of its target applications w.r.t. time-related aspects like throughput, resource utilization, experienced latencies, and congestion patterns. To support this type of analysis, the untimed DES behavioral models are extended to their *timed* versions. This extension takes place by endowing the original untimed models with additional attributes that characterize the experienced delays between the activation of an event and its execution (provided that it is not preempted by some other conflicting event). Timed models are further classified by the extent and the nature of the randomness that is captured by them. A basic such categorization is between deterministic models, where the aforementioned delays take fixed values for every event and stochastic models which admit more general distributions. From an application standpoint, timed DES models connect DES theory to the multitude of applications that have been addressed by Dynamic Programming, Stochastic Control, and scheduling theory (Bertsekas 1995; Meyn 2008; Pinedo 2002). Also, in their most general definition, stochastic DES models provide the theoretical foundation of discrete-event simulation (Banks et al. 2009).

Similar to the case of behavioral DES theory, a practical concern that challenges the application of timed DES models for performance modeling, analysis, and control is the very large size of these models, even for fairly small systems. DES theory has tried to circumvent these computational challenges through the development of methodology that enables the assessment of the system performance, over a set of possible configurations, from the observation of its behavior and the resultant performance at a single configuration. The required observations can be obtained through simulation, and in many cases, they can be collected from a single realization – or sample path – of the observed behavior; but then, the considered methods can also be applied on the actual system, and thus, they become a tool for real-time optimization, adaptation, and learning. Collectively, the aforementioned methods define a "sensitivity"-based approach to DES performance modeling, analysis, and control (Cassandras and Lafortune 2008; ▶ Perturbation Analysis of Discrete Event Systems). Historically, DES sensitivity analysis originated in the early 1980s in an effort to address the performance

analysis and optimization of queueing systems w.r.t. certain structural parameters like the arrival and processing rates (Ho and Cao 1991). But the current theory addresses more general stochastic DES models that bring it closer to broader endeavors to support incremental optimization, approximation, and learning in the context of stochastic optimal control (Cao 2007). Some particular applications of DES sensitivity analysis for the performance optimization of production, telecom, and computing systems can be found in Cassandras and Strickland (1988), Cassandras (1994), Panayiotou and Cassandras (1999), Homem-de Mello et al. (1999), Fu and Xie (2002), and Santoso et al. (2005).

Another interesting development in time-based DES theory is the theory of (max,+) algebra (Baccelli et al. 1992). In its practical applications, this theory addresses the timed dynamics of systems that involve the synchronization of a number of concurrently executing processes with no conflicts among them, and it provides important structural results on the factors that determine the behavior of these systems in terms of the occurrence rates of various critical events and the experienced latencies among them. Motivational applications of (max,+) algebra can be traced in the design and control of telecommunication and data networks, manufacturing, and railway systems, and more recently the theory has found considerable practical application in the computation of repetitive/cyclical schedules that seek to optimize the throughput rate of automated robotic cells and of the cluster tools that are used in semiconductor manufacturing (Kim and Lee 2012; Lee 2008; Park et al. 1999).

Both sensitivity-based methods and the theory of (max,+) algebra that were discussed in the previous paragraphs are enabled by the explicit, formal modeling of the DES structure and behavior in the pursued performance analysis and control. This integrative modeling capability that is supported by DES theory also enables a profound analysis of the impact of the imposed behavioral-control policies upon the system performance and, thus, the pursuance of a more integrative approach to the synthesis of the behavioral and

the performance-oriented control policies that are necessary for any particular DES instantiation. This is a rather novel topic in the relevant DES literature, and some recent works in this direction can be found in Cao (2005), Li and Reveliotis (2013), Markovski and Su (2013), and David-Henriet et al. (2013).

The Roles of Abstraction and Fluidification

The notions of "abstraction" and "fluidification" play a significant role in mastering the complexity that arises in many DES applications. Furthermore, both of these concepts have an important role in defining the essence and the boundaries of DES-based modeling.

In general systems theory, abstraction can be broadly defined as the effort to develop simplified models for the considered dynamics that retain, however, adequate information to resolve the posed questions in an effective manner. In DES theory, abstraction has been pursued w.r.t. the modeling of both the timed and untimed behaviors, giving rise to hierarchical structures and models. A theory for hierarchical SC is presented in Wonham (2006), while some applications of hierarchical SC in the manufacturing domain are presented in Hill et al. (2010) and Schmidt (2012). In general, hierarchical SC relies on a "spatial" decomposition that tries to localize/encapsulate the plant behavior into a number of modules that interact through the communication structure that is defined by the hierarchy. On the other hand, when it comes to timed DES behavior and models, a popular approach seeks to define a hierarchical structure for the underlying decision-making process by taking advantage of the different time scales that correspond to the occurrence of the various event types. Some particular works that formalize and systematize this idea in the application context of production systems can be found in Gershwin (1994) and Sethi and Zhang (1994) and the references cited therein.

In fact, the DES models that have been employed in many application areas can be perceived themselves as abstractions of dynamics of a more continuous, time-driven nature, where the underlying plant undergoes some fundamental

(possibly structural) transition upon the occurrence of certain events that are defined either endogenously or exogenously w.r.t. these dynamics. The combined consideration of the discrete-event dynamics that are generated in the manner described above, with the continuous, time-driven dynamics that characterize the modalities of the underlying plant, has led to the extension of the original DES theory to the, so-called, hybrid systems theory. Hybrid systems theory is itself very rich, and it is covered in another section of this encyclopedia (see also ▶ Discrete Event Systems and Hybrid Systems, Connections Between). From an application standpoint, it increases substantially the relevance of the DES modeling framework and brings this framework to some new and exciting applications. Some of the most prominent applications concern the coordination of autonomous vehicles and robotic systems, and a nice anthology of works concerning the application of hybrid systems theory in this particular application area can be found in the IEEE Robotics and Automation magazine of September 2011. These works also reveal the strong affinity that exists between hybrid systems theory and the DES modeling paradigm. Along similar lines, hybrid systems theory underlies also the endeavors for the development of the Automated Highway Systems that have been explored for the support of the future urban traffic needs (Horowitz and Varaiya 2000). Finally, hybrid systems theory and its DES component have been explored more recently as potential tools for the formal modeling and analysis of the molecular dynamics that are studied by systems biology (Curry 2012).

Fluidification, on the other hand, is the effort to represent as continuous flows, dynamics that are essentially of discrete-event type, in order to alleviate the computational challenges that typically result from discreteness and its combinatorial nature. The resulting models serve as approximations of the original dynamics, frequently they have the formal structure of hybrid systems, and they define a basis for developing "relaxations" for the originally addressed problems. Usually, their justification is of an *ad hoc* nature, and the quality of the established

approximations is empirically assessed on the basis of the delivered results (by comparing these results to some "baseline" performance). There are, however, a number of cases where the relaxed fluid model has been shown to retain important behavioral attributes of its original counterpart (Dai 1995). Furthermore, some recent works have investigated more analytically the impact of the approximation that is introduced by these models on the quality of the delivered results (Wardi and Cassandras 2013). Some more works regarding the application of fluidification in the DES-theoretic modeling frameworks, and of the potential advantages that it brings in various application contexts, can be found in Srikant (2004), Meyn (2008), David and Alla (2005), and Cassandras and Yao (2013).

Summary and Future Directions

The discussion of the previous section has revealed the extensive application range and potential of DES theory and its ability to provide structured and rigorous solutions to complex and sometimes ill-defined problems. On the other hand, the same discussion has revealed the challenges that underlie many of the DES applications. The complexity that arises from the intricate and integrating nature of most DES models is perhaps the most prominent of these challenges. This complexity manifests itself in the involved computations, but also in the need for further infrastructure, in terms of modeling interfaces and computational tools, that will render DES theory more accessible to the practitioner.

The DES community is aware of this need, and the last few years have seen the development of a number of computational platforms that seek to implement and leverage the existing theory by connecting it to various application settings; indicatively, we mention DESUMA (Ricker et al. 2006), SUPREMICA (Akesson et al. 2006), and TCT (Feng and Wonham 2006) that support DES behavioral modeling, analysis, and control along the lines of DES SC theory, while the website entitled "The Petri Nets World" has an extensive database of tools that support modeling and

analysis through untimed and timed variations of the Petri net model. Model checking tools, like SMV and NuSpin, that are used for verification purposes are also important enablers for the practical application of DES theory, and, of course, there are a number of programming languages and platforms, like Arena, AutoMod, and Simio, that support discrete-event simulation. However, with the exception of the discrete-event-simulation software, which is a pretty mature industry, the rest of the aforementioned endeavors currently evolve primarily within the academic and the broader research community. Hence, a remaining challenge for the DES community is the strengthening and expansion of the aforementioned computational platforms to robust and user-friendly computational tools. The availability of such industrial-strength computational tools, combined with the development of a body of control engineers well-trained in DES theory, will be catalytic for bringing all the developments that were described in the earlier parts of this document even closer to the industrial practice.

Cross-References

▶ Diagnosis of Discrete Event Systems
▶ Discrete Event Systems and Hybrid Systems, Connections Between
▶ Modeling, Analysis, and Control with Petri Nets
▶ Models for Discrete Event Systems: An Overview
▶ Perturbation Analysis of Discrete Event Systems
▶ Supervisory Control of Discrete-Event Systems

Bibliography

Akesson K, Fabian M, Flordal H, Malik R (2006) SUPREMICA-an integrated environment for verification, synthesis and simulation of discrete event systems. In: Proceedings of the 8th international workshop on discrete event systems, Ann Arbor. IEEE, pp 384–385

Alenljung T, Lennartson B, Hosseini MN (2012) Sensor graphs for discrete event modeling applied to formal verification of PLCs. IEEE Trans Control Syst Technol 20:1506–1521

Andersson K, Richardsson J, Lennartson B, Fabian M (2010) Coordination of operations by relation extraction for manufacturing cell controllers. IEEE Trans Control Syst Technol 18: 414–429

Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity: an algebra for discrete event systems. Wiley, New York

Balemi S, Hoffmann GJ, Wong-Toi PG, Franklin GJ (1993) Supervisory control of a rapid thermal multiprocessor. IEEE Trans Autom Control 38:1040–1059

Banks J, Carson II JS, Nelson BL, Nicol DM (2009) Discrete-event system simulation, 5th edn. Prentice Hall, Upper Saddle

Bertsekas DP (1995) Dynamic programming and optimal control, vols 1, 2. Athena Scientific, Belmont

Brandin B (1996) The real-time supervisory control of an experimental manufacturing cell. IEEE Trans Robot Autom 12:1–14

Cao X-R (2005) Basic ideas for event-based optimization of Markov systems. Discret Event Syst Theory Appl 15:169–197

Cao X-R (2007) Stochastic learning and optimization: a sensitivity approach. Springer, New York

Cassandras CG (1994) Perturbation analysis and "rapid learning" in the control of manufacturing systems. In: Leondes CT (ed) Dynamics of discrete event systems, vol 51. Academic, Boston, pp 243–284

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York

Cassandras CG, Strickland SG (1988) Perturbation analytic methodologies for design and optimization of communication networks. IEEE J Sel Areas Commun 6:158–171

Cassandras CG, Yao C (2013) Hybrid models for the control and optimization of manufacturing systems. In: Campos J, Seatzu C, Xie X (eds) Formal methods in manufacturing. CRC/Taylor and Francis, Boca Raton

Chandra V, Huang Z, Kumar R (2003) Automated control synthesis for an assembly line using discrete event system theory. IEEE Trans Syst Man Cybern Part C 33:284–289

Curry JER (2012) Some perspectives and challenges in the (discrete) control of cellular systems. In: Proceedings of the WODES 2012, Guadalajar. IFAC, pp 1–3

Dai JG (1995) On positive Harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. Ann Appl Probab 5:49–77

David R, Alla H (1992) Petri nets and Grafcet: tools for modelling discrete event systems. Prentice-Hall, Upper Saddle

David R, Alla H (2005) Discrete, continuous and hybrid Petri nets. Springer, Berlin

David-Henriet X, Hardouin L, Raisch J, Cottenceau B (2013) Optimal control for timed event graphs under partial synchronization. In: Proceedings of the 52nd IEEE conference on decision and control, Florence. IEEE

Dubreil J, Darondeau P, Marchand H (2010) Supervisory control for opacity. IEEE Trans Autom Control 55:1089–1100

Endsley EW, Almeida EE, Tilbury DM (2006) Modular finite state machines: development and application to reconfigurable manufacturing cell controller generation. Control Eng Pract 14:1127–1142

Ezpeleta J, Colom JM, Martinez J (1995) A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Trans R&A 11:173–184

Feng L, Wonham WM (2006) TCT: a computation tool for supervisory control synthesis. In: Proceedings of the 8th international workshop on discrete event systems, Ann Arbor. IEEE, pp 388–389

Feng L, Wonham WM, Thiagarajan PS (2007) Designing communicating transaction processes by supervisory control theory. Formal Methods Syst Design 30:117–141

Fu M, Xie X (2002) Derivative estimation for buffer capacity of continuous transfer lines subject to operation-dependent failures. Discret Event Syst Theory Appl 12:447–469

Gershwin SB (1994) Manufacturing systems engineering. Prentice Hall, Englewood Cliffs

Giua A, Fanti MP, Seatzu C (2006) Monitor design for colored Petri nets: an application to deadlock prevention in railway networks. Control Eng Pract 10:1231–1247

Hill RC, Cury JER, de Queiroz MH, Tilbury DM, Lafortune S (2010) Multi-level hierarchical interface-based supervisory control. Automatica 46:1152–1164

Ho YC, Cao X-R (1991) Perturbation analysis of discrete event systems. Kluwer Academic, Boston

Homem-de Mello T, Shapiro A, Spearman ML (1999) Finding optimal material release times using simulation-based optimization. Manage Sci 45:86–102

Hopcroft JE, Ullman JD (1979) Introduction to automata theory, languages and computation. Addison-Wesley, Reading

Horowitz R, Varaiya P (2000) Control design of automated highway system. Proc IEEE 88: 913–925

Jeng M, Xie X, Peng MY (2002) Process nets with resources for manufacturing modeling and their analysis. IEEE Trans Robot Autom 18:875–889

Kim J-H, Lee T-E (2012) Feedback control design for cluster tools with wafer residency time constraints. In: IEEE conference on systems, man and cybernetics, Seoul. IEEE, pp 3063–3068

Kumar R, Takai S (2010) Decentralized prognosis of failures in discrete event systems. IEEE Trans Autom Control 55:48–59

Lee T-E (2008) A review of cluster tool scheduling and control for semiconductor manufacturing. In: Proceedings of the winter simulation conference, Miami. INFORMS, pp 1–6

Lewis RW (1998) Programming industrial control systems using IEC 1131-3. Technical report, The Institution of Electrical Engineers

Li M, Kumar R (2012) Model-based automatic test generation for Simulink/Stateflow using extended finite automaton. In: Proceedings of the CASE, Seoul. IEEE

Li R, Reveliotis S (2013) Performance optimization for a class of generalized stochastic Petri nets. In: Proceedings of the 52nd IEEE conference on decision and control, Florence. IEEE

Li Z, Zhou M, Wu N (2008) A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems. IEEE Trans Syst Man Cybern Part C 38:173–188

Liao H, Wang Y, Cho HK, Stanley J, Kelly T, Lafortune S, Mahlke S, Reveliotis S (2013) Concurrency bugs in multithreaded software: modeling and analysis using Petri nets. Discret Event Syst Theory Appl 23:157–195

Markovski J, Su R (2013) Towards optimal supervisory controller synthesis of stochastic nondeterministic discrete event systems. In: Proceedings of the 52nd IEEE conference on decision and control, Florence. IEEE

Meyn S (2008) Control techniques for complex networks. Cambridge University Press, Cambridge

Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77:541–580

Panayiotou CG, Cassandras CG (1999) Optimization of kanban-based manufacturing systems. Automatica 35:1521–1533

Park E, Tilbury DM, Khargonekar PP (1999) Modular logic controllers for machining systems: formal representations and performance analysis using Petri nets. IEEE Trans Robot Autom 15:1046–1061

Pinedo M (2002) Scheduling. Prentice Hall, Upper Saddle River

Reveliotis SA (2000) Conflict resolution in AGV systems. IIE Trans 32(7):647–659

Reveliotis SA (2005) Real-time management of resource allocation systems: a discrete event systems approach. Springer, New York

Reveliotis SA (2007) Algebraic deadlock avoidance policies for sequential resource allocation systems. In: Lahmar M (ed) Facility logistics: approaches and solutions to next generation challenges. Auerbach Publications, Boca Raton, pp 235–289

Reveliotis SA, Ferreira PM (1996) Deadlock avoidance policies for automated manufacturing cells. IEEE Trans Robot Autom 12:845–857

Reveliotis S, Nazeem A (2013) Deadlock avoidance policies for automated manufacturing systems using finite state automata. In: Campos J, Seatzu C, Xie X (eds) Formal methods in manufacturing. CRC/Taylor and Francis

Reveliotis S, Roszkowska E (2011) Conflict resolution in free-ranging multi-vehicle systems: a resource allocation paradigm. IEEE Trans Robot 27:283–296

Ricker L, Lafortune S, Gene S (2006) DESUMA: a tool integrating giddes and umdes. In: Proceedings of the 8th international workshop on discrete event systems, Ann Arbor. IEEE, pp 392–393

Saboori A, Hadjicostis CN (2012) Opacity-enforcing supervisory strategies via state estimator constructions. IEEE Trans Autom Control 57:1155–1165

Saboori A, Hadjicostis CN (2014) Current-state opacity formulations in probabilistic finite automata. IEEE Trans Autom Control 59:120–133

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1996) Failure diagnosis using disrcete

event models. IEEE Trans Control Syst Technol 4:105–124

Sampath R, Darabi H, Buy U, Liu J (2008) Control reconfiguration of discrete event systems with dynamic control specifications. IEEE Trans Autom Sci Eng 5:84–100

Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. Europ J Oper Res 167:96–115

Schmidt K (2012) Computation of supervisors for reconfigurable machine tools. In: Proceedings of the WODES 2012, Guadalajara. IFAC, pp 227–232

Sethi SP, Zhang Q (1994) Hierarchical decision making in stochastic manufacturing systems. Birkhäuser, Boston

Srikant R (2004) The mathematics of internet congestion control. Birkhäuser, Boston

Van der Aalst W (1997) Verification of workflow nets. In: Azema P, Balbo G (eds) Lecture notes in computer science, vol 1248. Springer, New York, pp 407–426

Wardi Y, Cassandras CG (2013) Approximate IPA: trading unbiasedness for simplicity. In: Proceedings of the 52nd IEEE conference on decision and control, Florence. IEEE

Wassyng A, Lawford M, Maibaum T (2011) Software certification experience in the Canadian nuclear industry: lessons for the future. In: EMSOFT'11, Taipei

Wightkin N, Guy U, Darabi H (2011) Formal modeling of sequential function charts with time Petri nets. IEEE Trans Control Syst Technol 19:455–464

Wonham WM (2006) Supervisory control of discrete event systems. Technical report ECE 1636F/1637S 2006-07, Electrical & Computer Eng., University of Toronto

Zhou M, Fanti MP (eds) (2004) Deadlock resolution in computer-integrated systems. Marcel Dekker, Singapore

ATM Modernization

▶ Air Traffic Management Modernization: Promise and Challenges

Auctions

Bruce Hajek
University of Illinois, Urbana, IL, USA

Abstract

Auctions are procedures for selling one or more items to one or more bidders. Auctions induce games among the bidders, so notions of equilibrium from game theory can be applied to auctions. Auction theory aims to characterize and compare the equilibrium outcomes for different types of auctions. Combinatorial auctions arise when multiple-related items are sold simultaneously.

Keywords

Auction; Combinatorial auction; Game theory

Introduction

Three commonly used types of auctions for the sale of a single item are the following:

- *First price auction:* Each bidder submits a bid one of the bidders submitting the maximum bid wins, and the payment for the item is the maximum bid. (In this context "wins" means receives the item, no matter what the payment.)
- *Second price auction* or *Vickrey auction*: Each bidder submits a bid, one of the bidders submitting the maximum bid wins, and the payment for the item is the second highest bid.
- *English auction:* The price for the item increases continuously or in some small increments, and bidders drop out at some points in time. Once all but one of the bidders has dropped out, the remaining bidder wins and the payment is the price at which the last of the other bidders dropped out.

A key goal of the theory of auctions is to predict how the bidders will bid, and predict the resulting outcomes of the auction: which bidder is the winner and what is the payment. For example, a seller may be interested in the expected payment (seller revenue). A seller may have the option to choose one auction format over another and be interested in revenue comparisons. Another item of interest is efficiency or social welfare. For sale of a single item, the outcome is efficient if the item is sold to the bidder with the highest value for the item. The book of V. Krishna (2002) provides an excellent introduction to the theory of auctions.

Auctions Versus Seller Mechanisms

An important class of mechanisms within the theory of mechanism design are seller mechanisms, which implement the sale of one or more items to one or more bidders. Some authors would consider all such mechanisms to be auctions, but the definition of auctions is often more narrowly interpreted, with auctions being the subclass of seller mechanisms which do not depend on the fine details of the set of bidders. The rules of the three types of auction mentioned above do not depend on fine details of the bidders, such as the number of bidders or statistical information about how valuable the item is to particular bidders. In contrast, designing a procedure to sell an item to a known set of bidders under specific statistical assumptions about the bidders' preferences in order to maximize the expected revenue (as in Myerson (1981)) would be considered a problem of mechanism design, which is outside the more narrowly defined scope of auctions. The narrower definition of auctions was championed by R. Wilson (1987). An article on ▶ Mechanism Design appears in this encyclopedia.

Equilibrium Strategies in Auctions

An auction induces a noncooperative game among the bidders, and a commonly used predictor of the outcome of the auction is an equilibrium of the game, such as a Nash or Bayes-Nash equilibrium. For a risk neutral bidder i with value x_i for the item, if the bidder wins and the payment is M_i, the payoff of the bidder is $x_i - M_i$. If the bidder does not win, the payoff of the bidder is zero. If, instead, the bidder is risk averse with risk aversion measured by an increasing utility function u_i, the payoff of the bidder would be $u_i(x_i - M_i)$ if the bidder wins and $u_i(0)$ if the bidder does not win.

The second price auction format is characterized by simplicity of the bidding strategies. If bidder i knows the value x_i of the item to himself, then for the second price auction format, a weakly dominant strategy for the bidder is to truthfully report x_i as his bid for the item. Indeed, if y_i is the highest bid of the other bidders, the payoff of bidder i is $u_i(x_i - y_i)$ if he wins and $u_i(0)$ if he does not win. Thus, bidder i would prefer to win whenever $u_i(x_i - y_i) > u_i(0)$ and not win whenever $u_i(x_i - y_i) < u_i(0)$. That is precisely what happens if bidder i bids x_i, no matter what the bids of the other bidders are. That is, bidding x_i is a weakly dominant strategy for bidder i.

Nash equilibrium can be found for the other types of auctions under a model with incomplete information, in which the type of each bidder i is equal to the value of the object to the bidder and is modeled as a random variable X_i with a density function f_i supported by some interval $[a_i, b_i]$. A simple case is that the bidders are all risk neutral, the densities are all equal to some fixed density f, and the X_i's are mutually independent. The English auction in this context is equivalent to the second price auction: in an English auction, dropping out when the price reaches his true value is a weakly dominant strategy for a bidder, and for the weakly dominant strategy equilibrium, the outcome of the auction is the same as for the second price auction. For the first price auction in this symmetric case, there exists a symmetric Bayesian equilibrium. It corresponds to all bidders using the bidding function β (so the bid of bidder i is $\beta(X_i)$), where β is given by $\beta(x) = E[Y_1|Y_1 \leq x]$. The expected revenue to the seller in this case is $E[Y_1|Y_1 < X_1]$, which is the same as the expected revenue for the second price auction and English auction.

Equilibrium for Auctions with Interdependent Valuations

Seminal work of Milgrom and Weber (1982) addresses the performance of the above three auction formats in case the bidders do not know the value of the item, but each bidder i has a private signal X_i about the value V_i of the item to bidder i. The values and signals $(X_1, \ldots X_n, V_1, \ldots, V_n)$ can be interdependent. Under the assumption of invariance of the joint distribution of $(X_1, \ldots X_n, V_1, \ldots, V_n)$ under permutation of the bidders and a strong form of positive correlation of the random

variables $(X_1, \ldots X_n, V_1, \ldots, V_n)$ (see Milgrom and Weber 1982 or Krishna 2002 for details), a symmetric Bayes-Nash equilibrium is identified for each of the three auction formats mentioned above, and the expected revenues for the three auction formats are shown to satisfy the ordering $R^{(\text{first price})} \leq R^{(\text{second price})} \leq R^{(\text{English})}$. A significant extension of the theory of Milgrom and Weber due to DeMarzo et al. (2005) is the theory of security-bid auctions in which bidders compete to buy an asset and the final payment is determined by a contract involving the value of the asset as revealed after the auction.

Combinatorial Auctions

Combinatorial auctions implement the simultaneous sale of multiple items. A simple version is the simultaneous ascending price auction with activity constraints (Cramton 2006; Milgrom 2004). Such an auction procedure was originally proposed by Preston, McAfee, Paul Milgrom, and Robert Wilson for the US FCC wireless spectrum auction in 1994 and was used for the vast majority of spectrum auctions worldwide since then Cramton (2013). The auction proceeds in rounds. In each round a minimum price is set for each item, with the minimum prices for the initial round being reserve prices set by the seller. A given bidder may place a bid on an item in a given round such that the bid is greater than or equal to the minimum price for the item. If one or more bidders bid on an item in a round, a provisional winner of the item is selected from among the bidders with the highest bid for the item in the round, with the new provisional price being the highest bid. The minimum price for the item is increased 10 % (or some other small percentage) above the new provisional price. Once there is a round with no bids, the set of provisional winners is identified. Often constraints are placed on the bidders in the form of *activity rules.* An activity rule requires a bidder to maintain a history of bidding in order to continue bidding, so as to prevent bidders from not bidding in early rounds and bidding aggressively in later rounds. The motivation for activity rules is to promote *price*

discovery to help bidders select the packages (or bundles) of items most suitable for them to buy. A key is that complementarities may exist among the items for a given bidder. Complementarity means that a bidder may place a significantly higher value on a bundle of items than the sum of values the bidder would place on the items individually. Complementarities lead to the *exposure problem*, which occurs when a bidder wins only a subset of items of a desired bundle at a price which is significantly higher than the price paid. For example, a customer might place a high value on a particular pair of shoes, but little value on a single shoe alone.

A variation of simultaneous ascending price auctions for combinatorial auctions is auctions with package bidding (see, e.g., Ausubel and Milgrom 2002; Cramton 2013). A bidder will either win a package of items he bid for or no items, thereby eliminating the exposure problem. For example, in simultaneous clock auctions with package bidding, the price for each item increases according to a fixed schedule (the clock), and bidders report the packages of items they would prefer to purchase for the given prices. The price for a given item stops increasing when the number of bidders for that item drops to zero or one, and the clock phase of the auction is complete when the number of bidders for every item is zero or one. Following the clock phase, bidders can submit additional bids for packages of items. With the inputs from bidders acquired during the clock phase and supplemental bid phase, the auctioneer then runs a winner determination algorithm to select a set of bids for non-overlapping packages that maximizes the sum of the bids. This winner determination problem is NP hard, but is computationally feasible using integer programming or dynamic programming methods for moderate numbers of items (perhaps up to 30). In addition, the vector of payments charged to the winners is determined by a two-step process. First, the (generalized) Vickrey price for each bidder is determined, which is defined to be the minimum the bidder would have had to bid in order to be a winner. Secondly, the vector of Vickrey prices is projected onto the core of the reported prices. The second step insures that no coalition

consisting of a set of bidders and the seller can achieve a higher sum of payoffs (calculated using the bids received) for some different selection of winners than the coalition received under the outcome of the auction. While this is a promising family of auctions, the projection to the core introduces some incentive for bidders to deviate from truthful reporting, and much remains to be understood about such auctions.

Summary and Future Directions

Auction theory provides a good understanding of the outcomes of the standard auctions for the sale of a single item. Recently emerging auctions, such as for the generation and consumption of electrical power, and for selection of online advertisements, are challenging to analyze and comprise a direction for future research. Much remains to be understood in the theory of combinatorial auctions, such as the degree of incentive compatibility offered by core projecting auctions.

Cross-References

▶ Game Theory: Historical Overview

Bibliography

Ausubel LM, Milgrom PR (2002) Ascending auctions with package bidding. BE J Theor Econ 1(1):Article 1
Cramton P (2006) Simultaneous ascending auctions. In: Cramton P, Shoham Y, Steinberg R (eds) Combinatorial auctions, chapter 4. MIT, Cambridge, pp 99–114
Cramton P (2013) Spectrum auction design. Rev Ind Organ 42(4):161–190
DeMarzo PM, Kremer I, Skrzypacz A (2005) Bidding with securities: auctions and security bidding with securities: auctions and security design. Am Econ Rev 95(4):936–959
Krishna V (2002) Auction theory. Academic, San Diego
Milgrom PR (2004) Putting auction theory to work. Cambridge University Press, Cambridge/ New York
Milgrom PR, Weber RJ (1982) A theory of auctions and competitive bidding. Econometrica 50(5):1089–1122
Myerson R (1981) Optimal auction design. Math Oper Res 6(1):58–73
Wilson R (1987) Game theoretic analysis of trading processes. In: Bewley T (ed) Advances in economic theory. Cambridge University Press, Cambridge/New York

Autotuning

Tore Hägglund
Lund University, Lund, Sweden

Abstract

Autotuning, or automatic tuning, means that the controller is tuned automatically. Autotuning is normally applied to PID controllers, but the technique can also be used to initialize more advanced controllers. The main approaches to autotuning are based on step response analysis or frequency response analysis obtained using relay feedback. Autotuning has been well received in industry, and today most distributed control systems have some kind of autotuning technique.

Keywords

Automatic tuning; Gain scheduling; PID control; Process control; Proportional-integral-derivative control; Relay feedback

Background

In the late 1970s and early 1980s, there was a quite rapid change of controller implementation in process control. The analog controllers were replaced by computer-based controllers and distributed control systems. The functionality of the new controllers was often more or less a copy of the old analog equipment, but new functions that utilized the computer implementation were gradually introduced. One of the first functions of this type was autotuning. Autotuning is a method to tune the controllers, normally PID controllers, automatically.

What Is Autotuning?

A PID controller in its basic form has the structure

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{d}{dt}e(t)\right),$$

where u is the controller output and $e = y_{sp} - y$ is the control error, where y_{sp} is the setpoint and y is the process output. There are three parameters in the controller, gain K, integral time T_i, and derivative time T_d. These parameters have to be set by the user. Their values are dependent of the process dynamics and the specifications of the control loop.

A process control plant may have thousands of control loops, which means that maintaining high-performance controller tuning can be very time consuming. This was the main reason why procedures for automatic tuning were installed so rapidly in the computer-based controllers.

When a controller is to be tuned, the following steps are normally performed by the user:
1. To determine the process dynamics, a minor disturbance is injected by changing the control signal.
2. By studying the response in the process output, the process dynamics can be determined, i.e., a process model is derived.
3. The controller parameters are finally determined based on the process model and the specifications.

Autotuning means simply that these three steps are performed automatically. Instead of having a human to perform these tasks, they are performed automatically on demand from the user. Ideally, the autotuning should be fully automatic, which means that no information about the process dynamics is required from the user.

Automatic tuning can be performed in many ways. The process disturbance can take different forms, e.g., in the form of step changes or some kind of oscillatory excitation. The model obtained can be more or less accurate. There are also many ways to tune the controller based on the process model.

Here, we will discuss two main approaches for autotuning, namely, those that are based on step response analysis and those that are based on frequency response analysis.

Methods Based on Step Response Analysis

Most methods for automatic tuning of PID controllers are based on step response analysis. When the operator wishes to tune the controller, an open-loop step response experiment is performed. A process model is then obtained from the step response, and controller parameters are determined. This is usually done using simple formulas or look-up tables.

The most common process model used for PID controller tuning based on step response experiments is the first-order plus dead-time model

$$G(s) = \frac{K_p}{1 + sT}e^{-sL}$$

where K_p is the static gain, T is the apparent time constant, and L is the apparent dead time. These three parameters can be obtained from a step response experiment according to Fig. 1.

Static gain K_p is given by the ratio between the steady-state change in process output and the magnitude of the control signal step, $K_p = \Delta y/\Delta u$. Dead-time L is determined from the time elapsed from the step change to the intersection of the largest slope of the process output with the level of the process output before the step change. Finally, time constant T is the time when the process output has reached 63 % of its final value, subtracted by L.
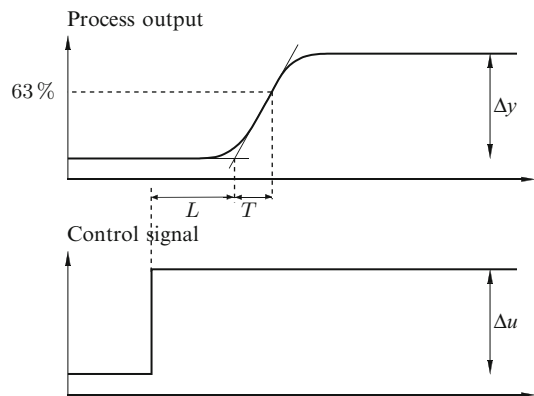

Autotuning, Fig. 1 Determination of K_p, L, and T from a step response experiment

The greatest difficulty in carrying out tuning automatically is in selecting the amplitude of the step. The user naturally wants the disturbance to be as small as possible so that the process is not disturbed more than necessary. On the other hand, it is easier to determine the process model if the disturbance is large. The result of this dilemma is usually that the user has to decide how large the step in the control signal should be. Another problem is to determine when the step response has reached its final value.

Methods Based on Frequency Response Analysis

Frequency-domain characteristics of the process can be obtained by adding sinusoidals to the control signal, but without knowing the frequency response of the process, the interesting frequency range and acceptable amplitudes are not known. A method that automatically provides a relevant frequency response can be determined from experiments with relay feedback according to Fig. 2. Notice that there is a switch that selects either relay feedback or ordinary PID feedback. When it is desired to tune the system, the PID function is disconnected and the system is connected to relay feedback control. Relay feedback control is the same as on/off control, but where the on and off levels are carefully chosen and not 0 and 100 %. The relay feedback makes the control loop oscillate. The period and the amplitude of the oscillation is determined when steady-state oscillation is obtained. This gives the ultimate period and the ultimate gain. The parameters of a PID controller can then be determined from these values. The PID controller is then automatically switched in again, and the control is executed with the new PID parameters.

For large classes of processes, relay feedback gives an oscillation with period close to the ultimate frequency ω_u, as shown in Fig. 3, where the control signal is a square wave and the process output is close to a sinusoid. The gain of the transfer function at this frequency is also easy to obtain from amplitude measurements.
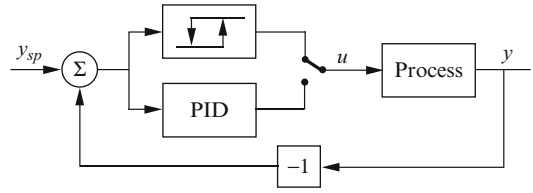
Autotuning, Fig. 2 The relay autotuner. In the tuning mode the process is connected to relay feedback

Describing function analysis can be used to determine the process characteristics. The describing function of a relay with hysteresis is

$$N(a) = \frac{4d}{\pi a} \left(\sqrt{1 - \left(\frac{\epsilon}{a}\right)^2} - i\,\frac{\epsilon}{a} \right)$$

where d is the relay amplitude, ϵ the relay hysteresis, and a the amplitude of the input signal. The negative inverse of this describing function is a straight line parallel to the real axis; see Fig. 4. The oscillation corresponds to the point where the negative inverse describing function crosses the Nyquist curve of the process, i.e., where

$$G(i\omega) = -\frac{1}{N(a)}$$

Since $N(a)$ is known, $G(i\omega)$ can be determined from the amplitude a and the frequency ω of the oscillation.

Notice that the relay experiment is easily automated. There is often an initialization phase where the noise level in the process output is determined during a short period of time. The noise level is used to determine the relay hysteresis and a desired oscillation amplitude in the process output. After this initialization phase, the relay function is introduced. Since the amplitude of the oscillation is proportional to the relay output, it is easy to control it by adjusting the relay output.

Different Adaptive Techniques

In the late 1970s, at the same time as autotuning procedures were developed and implemented in industrial controllers, there was a large academic interest in adaptive control. These two concepts
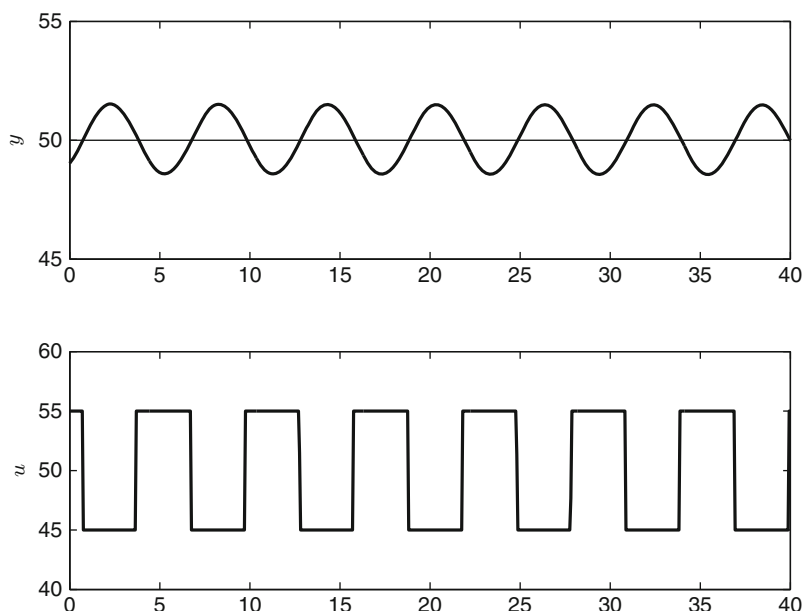
Autotuning, Fig. 3 Process output y and control signal u during relay feedback
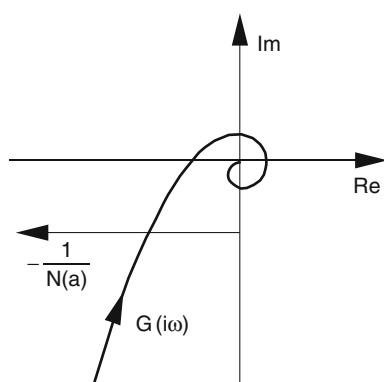
Autotuning, Fig. 4 The negative inverse describing function of a relay with hysteresis $-1/N(a)$ and a Nyquist curve $G(i\omega)$

are often mixed up with each other. Autotuning is sometimes called tuning on demand. An identification experiment is performed, controller parameters are determined, and the controller is then run with fixed parameters. An adaptive controller is, however, a controller where the controller parameters are adjusted online based on information from routine data. Automatic tuning and adaptive control have, however, one thing in common, namely, that they are methods to adapt the controller parameters to the actual process

dynamics. Therefore, they are both called *adaptive techniques*.

There is a third adaptive technique, namely, gain scheduling. Gain scheduling is a system where controller parameters are changed depending on measured operating conditions. The scheduling variable can, for instance, be the measurement signal, controller output, or an external signal. For historical reasons the word gain scheduling is used even if other parameters like integral time or derivative time are changed. Gain scheduling is a very effective way of controlling systems whose dynamics change with the operating conditions. Automatic tuning has made it possible to generate gain schedules automatically.

Although research on adaptive techniques has almost exclusively focused on adaptation, experience has shown that autotuning and gain scheduling have much wider industrial applicability. Figure 5 illustrates the appropriate use of the different techniques.

Controller performance is the first issue to consider. If requirements are modest, a controller with constant parameters and conservative tuning can be used. Other solutions should be considered when higher performance is required.

Autotuning, Fig. 5 When to use different adaptive techniques
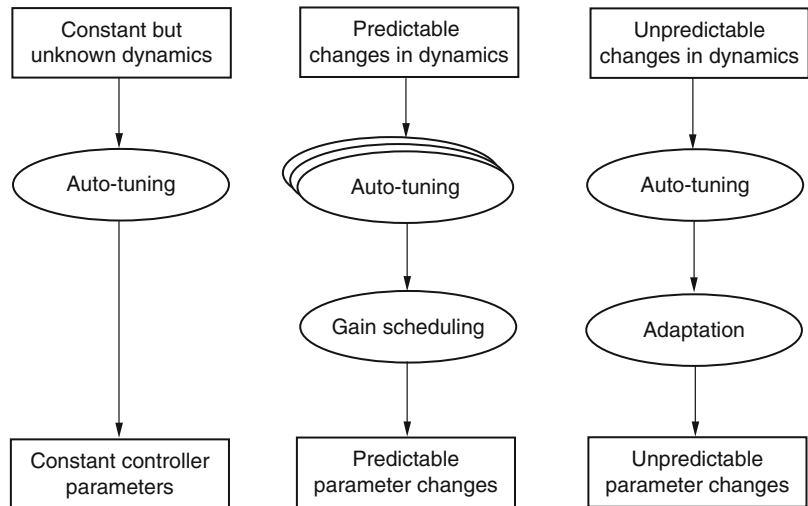
If the process dynamics are constant, a controller with constant parameters should be used. The parameters of the controller can be obtained by autotuning.

If the process dynamics or the character of the disturbances are changing, it is useful to compensate for these changes by changing the controller. If the variations can be predicted from measured signals, gain scheduling should be used since it is simpler and gives superior and more robust performance than continuous adaptation. Typical examples are variations caused by nonlinearities in the control loop. Autotuning can be used to build up the gain schedules automatically.

There are also cases where the variations in process dynamics are not predictable. Typical examples are changes due to unmeasurable variations in raw material, wear, fouling, etc. These variations cannot be handled by gain scheduling but must be dealt with by adaptation. An autotuning procedure is often used to initialize the adaptive controller. It is then sometimes called pre-tuning or initial tuning.

To summarize, autotuning is a key component in all adaptive techniques and a prerequisite for their use in practice.

Industrial Products

Commercial PID controllers with adaptive techniques have been available since the beginning of the late 1970s, both in single-station controllers and in distributed control systems.

Two important, but distinct, applications of PID autotuners are *temperature controllers* and *process controllers*. Temperature controllers are primarily designed for temperature control, whereas process controllers are supposed to work for a wide range of control loops in the process industry such as flow, pressure, level, temperature, and concentration control loops. Automatic tuning is easier to implement in temperature controllers, since most temperature control loops have several common features. This is the main reason why automatic tuning was introduced more rapidly in these controllers.

Since the processes that are controlled with process controllers may have large differences in their dynamics, tuning becomes more difficult compared to the pure temperature control loops.

Automatic tuning can also be performed by external devices which are connected to the control loop during the tuning phase. Since these devices are supposed to work together with controllers from different manufacturers, they must

be provided with quite a lot of information about the controller structure and parameterization in order to provide appropriate controller parameters. Such information includes signal ranges, controller structure (series or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time).

Summary and Future Directions

Most of the autotuning methods that are available in industrial products today were developed about 30 years ago, when computer-based controllers started to appear. These autotuners are often based on simple models and simple tuning rules. With the computer power available today, and the increased knowledge about PID controller design, there is a potential for improving the autotuners, and more efficient autotuners will probably appear in industrial products quite soon.

Cross-References

▶ Adaptive Control, Overview
▶ PID Control

Bibliography

Åström KJ, Hägglund T (1995) PID controllers: theory, design, and tuning. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park
Åström KJ, Hägglund T (2005) Advanced PID control. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park
Vilanova R, Visioli A (eds) (2012) PID control in the third millennium. Springer, Dordrecht
Visioli A (2006) Practical PID control. Springer, London
Yu C-C (2006) Autotuning of PID controllers – a relay feedback approach. Springer, London

Averaging Algorithms and Consensus

Wei Ren
Department of Electrical Engineering,
University of California, Riverside, CA, USA

A

Abstract

In this article, we overview averaging algorithms and consensus in the context of distributed coordination and control of networked systems. The two subjects are closely related but not identical. Distributed consensus means that a team of agents reaches an agreement on certain variables of interest by interacting with their neighbors. Distributed averaging aims at computing the average of certain variables of interest among multiple agents by local communication. Hence averaging can be treated as a special case of consensus – average consensus. For distributed consensus, we introduce distributed algorithms for agents with single-integrator, general linear, and nonlinear dynamics. For distributed averaging, we introduce static and dynamic averaging algorithms. The former is useful for computing the average of initial conditions (or constant signals), while the latter is useful for computing the average of time-varying signals. Future research directions are also discussed.

Keywords

Cooperative control; Coordination; Distributed control; Multi-agent systems; Networked systems

Introduction

In the area of control of networked systems, low cost, high adaptivity and scalability, great robustness, and easy maintenance are critical factors.

To achieve these factors, distributed coordination and control algorithms that rely on only local interaction between neighboring agents to achieve collective group behavior are more favorable than centralized ones. In this article, we overview averaging algorithms and consensus in the context of distributed coordination and control of networked systems.

Distributed consensus means that a team of agents reaches an agreement on certain variables of interest by interacting with their neighbors. A consensus algorithm is an update law that drives the variables of interest of all agents in the network to converge to a common value (Jadbabaie et al. 2003; Olfati-Saber et al. 2007; Ren and Beard 2008). Examples of the variables of interest include a local representation of the center and shape of a formation, the rendezvous time, the length of a perimeter being monitored, the direction of motion for a multi-vehicle swarm, and the probability that a target has been identified. Consensus algorithms have applications in rendezvous, formation control, flocking, attitude alignment, and sensor networks (Bai et al. 2011a; Bullo et al. 2009; Mesbahi and Egerstedt 2010; Qu 2009; Ren and Cao 2011). Distributed averaging algorithms aim at computing the average of certain variables of interest among multiple agents by local communication. Distributed averaging finds applications in distributed computing, distributed signal processing, and distributed optimization (Tsitsiklis et al. 1986). Hence the variables of interest are dependent on the applications (e.g., a sensor measurement or a network quantity). Consensus and averaging algorithms are closely connected and yet nonidentical. When all agents are able to compute the average, they essentially reach a consensus, the so-called average consensus. On the other hand, when the agents reach a consensus, the consensus value might or might not be the average value.

Graph Theory Notations. Suppose that there are n agents in a network. A *network topology* (equivalently, *graph*) $\mathcal{G}$ consisting of a node set $\mathcal{V} = \{1, \ldots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ will be used to model *interaction* (communication or sensing) between the n agents. An *edge* (i, j) in

a *directed graph* denotes that agent j can obtain information from agent i, but not necessarily vice versa. In contrast, an edge (i, j) in an *undirected graph* denotes that agents i and j can obtain information from each other. Agent j is a *(in-) neighbor* of agent i if $(j, i) \in \mathcal{E}$. Let $\mathcal{N}_i$ denote the neighbor set of agent i. We assume that $i \in \mathcal{N}_i$. A *directed path* is a sequence of edges in a directed graph of the form $(i_1, i_2), (i_2, i_3), \ldots$, where $i_j \in \mathcal{V}$. An *undirected path* in an undirected graph is defined analogously. A directed graph is *strongly connected* if there is a directed path from every agent to every other agent. An undirected graph is *connected* if there is an undirected path between every pair of distinct agents. A directed graph *has a directed spanning tree* if there exists at least one agent that has directed paths to all other agents. For example, Fig. 1 shows a directed graph that has a directed spanning but is not strongly connected. The *adjacency matrix* $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ associated with $\mathcal{G}$ is defined such that a_{ij} (the weight of edge (j, i)) is positive if agent j is a neighbor of agent i while $a_{ij} = 0$ otherwise. The (nonsymmetric) *Laplacian matrix* (Agaev and Chebotarev 2005) $\mathcal{L} = [\ell_{ij}] \in \mathbb{R}^{n \times n}$ associated with $\mathcal{A}$ and hence $\mathcal{G}$ is defined as $\ell_{ii} = \sum_{j \neq i} a_{ij}$ and $\ell_{ij} = -a_{ij}$ for all $i \neq j$. For an undirected graph, we assume that $a_{ij} = a_{ji}$. A graph is *balanced* if for every agent the total edge weights of its incoming links is equal to the total edge weights of its outgoing links ($\sum_{j=1}^{n} a_{ij} = \sum_{j=1}^{n} a_{ji}$ for all i).
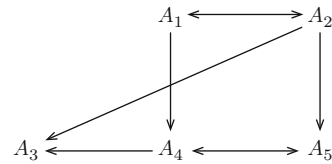
Averaging Algorithms and Consensus, Fig. 1 A directed graph that characterizes the interaction among five agents, where A_i, $i = 1, \ldots, 5$, denotes agent i. An *arrow* from agent j to agent i indicates that agent i receives information from agent j. The directed graph has a directed spanning tree but is not strongly connected. Here both agents 1 and 2 have directed paths to all other agents

Consensus

Consensus has a long history in management science, statistical physics, and distributed computing and finds recent interests in distributed control. While in the area of distributed control of networked systems the term *consensus* was initially more or less dominantly referred to the case of a continuous-time version of a distributed linear averaging algorithm, such a term has been broadened to a great extent later on. Related problems to consensus include synchronization, agreement, and rendezvous. The study of consensus can be categorized in various manners. For example, in terms of the final consensus value, the agents could reach a consensus on the average, a weighted average, the maximum value, the minimum value, or a general function of their initial conditions, or even a (changing) state that serves as a reference. A consensus algorithm could be linear or nonlinear. Consensus algorithms can be designed for agents with linear or nonlinear dynamics. As the agent dynamics become more complicated, so do the algorithm design and analysis. Numerous issues are also involved in consensus such as network topologies (fixed vs. switching, deterministic vs. random, directed vs. undirected, asynchronous vs. synchronous), time delay, quantization, optimality, sampling effects, and convergence speed. For example, in real applications, due to nonuniform communication/sensing ranges or limited field of view of sensors, the network topology could be directed rather than undirected. Also due to unreliable communication/sensing and limited communication/sensing ranges, the network topology could be switching rather than fixed.

Consensus for Agents
with Single-Integrator Dynamics

We start with a fundamental consensus algorithm for agents with single-integrator dynamics. The results in this section follow from Jadbabaie et al. (2003), Olfati-Saber et al. (2007), Ren and Beard (2008), Moreau (2005), and Agaev and Chebotarev (2000). Consider agents with single-integrator dynamics

$$\dot{x}_i(t) = u_i(t), \quad i = 1, \ldots, n, \qquad (1)$$

where x_i is the state and u_i is the control input. A common consensus algorithm for (1) is

$$u_i(t) = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)[x_j(t) - x_i(t)], \quad (2)$$

where $\mathcal{N}_i(t)$ is the neighbor set of agent i at time t and $a_{ij}(t)$ is the (i, j) entry of the adjacency matrix $\mathcal{A}$ of the graph $\mathcal{G}$ at time t. A consequence of (2) is that the state $x_i(t)$ of agent i is driven toward the states of its neighbors or equivalently toward the weighted average of its neighbors' states. The closed-loop system of (1) using (2) can be written in matrix form as

$$\dot{x}(t) = -\mathcal{L}(t)x(t), \qquad (3)$$

where x is a column stack vector of all x_i and $\mathcal{L}$ is the Laplacian matrix. Consensus is *reached* if for all initial states, the agents' states eventually become identical. That is, for all $x_i(0)$, $\|x_i(t) - x_j(t)\|$ approaches zero eventually.

The properties of the Laplacian matrix $\mathcal{L}$ play an important role in the analysis of the closed-loop system (3). When the graph $\mathcal{G}$ (and hence the associated Laplacian matrix $\mathcal{L}$) is fixed, (3) can be analyzed by studying the eigenvalues and eigenvectors of $\mathcal{L}$. Due to its special structure, for any graph $\mathcal{G}$, the associated Laplacian matrix $\mathcal{L}$ has at least one zero eigenvalue with an associated right eigenvector $\mathbf{1}$ (column vector of all ones) and all other eigenvalues have positive real parts. To ensure consensus, it is equivalent to ensure that $\mathcal{L}$ has a simple zero eigenvalue. It can be shown that the following three statements are equivalent: (i) the agents reach a consensus exponentially for arbitrary initial states; (ii) the graph $\mathcal{G}$ has a directed spanning tree; and (iii) the Laplacian matrix $\mathcal{L}$ has a simple zero eigenvalue with an associated right eigenvector $\mathbf{1}$ and all other eigenvalues have positive real parts. When consensus is reached, the final consensus value is a weighted average of the initial states of those agents that have directed paths to all other agents (see Fig. 2 for an illustration).

Averaging Algorithms and Consensus, Fig. 2 Consensus for five agents using the algorithm (2) for (1). Here the graph $\mathcal{G}$ is given by Fig. 1. The initial states are chosen as $x_i(0) = 2i$, where $i = 1, \ldots, 5$. Consensus is reached as $\mathcal{G}$ has a directed spanning tree. The final consensus value is a weighted average of the initial states of agents 1 and 2
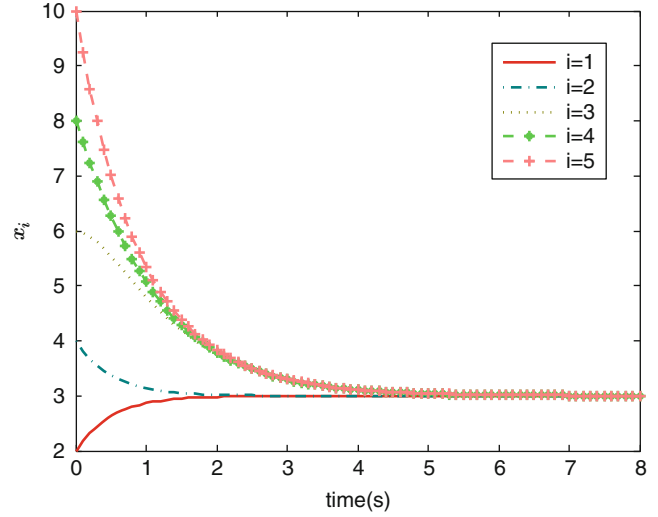
When the graph $\mathcal{G}(t)$ is switching at time instants $t_0, t_1, \ldots$, the solution to the closed-loop system (3) is given by $x(t) = \Phi(t, 0)x(0)$, where $\Phi(t, 0)$ is the transition matrix corresponding to $-\mathcal{L}(t)$. Consensus is reached if $\Phi(t, 0)$ eventually converges to a matrix with identical rows. Here $\Phi(t, 0) = \Phi(t, t_k)\Phi(t_k, t_{k-1}) \cdots \Phi(t_1, 0)$, where $\Phi(t_k, t_{k-1})$ is the transition matrix corresponding to $\mathcal{L}(t)$ at time interval $[t_{k-1}, t_k]$. It turns out that each transition matrix is a *row-stochastic* matrix with positive diagonal entries. A square matrix is row stochastic if all its entries are nonnegative and all of its row sums are one. The consensus convergence can be analyzed by studying the product of row-stochastic matrices. Another analysis technique is a Lyapunov approach (e.g., $\max x_i - \min x_i$). It can be shown that the agents' states reach a consensus if there exists an infinite sequence of contiguous, uniformly bounded time intervals, with the property that across each such interval, the union of the graphs $\mathcal{G}(t)$ has a directed spanning tree. That is, across each such interval, there exists at least one agent that can directly or indirectly influence all other agents. It is also possible to achieve certain nice features by designing nonlinear consensus algorithms of the form $u_i(t) = \sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)\psi[x_j(t) - x_i(t)]$, where $\psi(\cdot)$ is a nonlinear function satisfying certain properties. One example is a continuous nondecreasing odd function. For example, a saturation type function could be introduced to

account for actuator saturation and a signum type function could be introduced to achieve finite-time convergence.

As shown above, for single-integrator dynamics, the consensus convergence is determined entirely by the network topologies. The primary reason is that the single-integrator dynamics are internally stable. However, when more complicated agent dynamics are involved, the consensus algorithm design and analysis become more complicated. On one hand, whether the graph is undirected (respectively, switching) or not has significant influence on the complexity of the consensus analysis. On the other hand, not only the network topology but also the agent dynamics themselves and the parameters in the consensus algorithm play important roles. Next we introduce consensus for agents with general linear and nonlinear dynamics.

Consensus for Agents with General Linear Dynamics

In some circumstances, it is relevant to deal with agents with general linear dynamics, which can also be regarded as linearized models of certain nonlinear dynamics. The results in this section follow from Li et al. (2010). Consider agents with general linear dynamics

$$\dot{x}_i = Ax_i + Bu_i, \quad y_i = Cx_i, \qquad (4)$$

where $x_i \in \mathbb{R}^m$, $u_i \in \mathbb{R}^p$, and $y_i \in \mathbb{R}^q$ are, respectively, the state, the control input, and the output of agent i and A, B, C are constant matrices with compatible dimensions.

When each agent has access to the relative states between itself and its neighbors, a distributed static consensus algorithm is designed for (4) as

$$u_i = cK \sum_{j \in \mathcal{N}_i} a_{ij}(x_i - x_j), \qquad (5)$$

where $c > 0$ is a coupling gain, $K \in \mathbb{R}^{p \times m}$ is the feedback gain matrix, and $\mathcal{N}_i$ and a_{ij} are defined as in (2). It can be shown that if the graph $\mathcal{G}$ has a directed spanning tree, consensus is reached using (5) for (4) if and only if all the matrices $A + c\lambda_i(\mathcal{L})BK$, where $\lambda_i(\mathcal{L}) \neq 0$ are Hurwitz. Here $\lambda_i(\mathcal{L})$ denotes the ith eigenvalue of the Laplacian matrix $\mathcal{L}$. A necessary condition for reaching a consensus is that the pair (A, B) is stabilizable. The consensus algorithm (5) can be designed via two steps:

(a) Solve the linear matrix inequality $A^T P + PA - 2BB^T < 0$ to get a positive-definite solution P. Then let the feedback gain matrix $K = -B^T P^{-1}$.
(b) Select the coupling strength c larger than the threshold value $1/ \min_{\lambda_i(\mathcal{L}) \neq 0} \mathrm{Re}[\lambda_i(\mathcal{L})]$, where $\mathrm{Re}(\cdot)$ denotes the real part.

Note that here the threshold value depends on the eigenvalues of the Laplacian matrix, which is in some sense global information. To overcome such a limitation, it is possible to introduce adaptive gains in the algorithm design. The gains could be updated dynamically using local information.

When the relative states between each agent and its neighbors are not available, one is motivated to make use of the output information and employ observer-based design to estimate the relative states. An observer-type consensus algorithm is designed for (4) as

$$\dot{v}_i = (A + BF)v_i + cL \sum_{j \in \mathcal{N}_i} a_{ij}[C(v_i - v_j) - (y_i - y_j)],$$
$$u_i = Fv_i, \quad i = 1, \cdots, n, \qquad (6)$$

where $v_i \in \mathbb{R}^m$ are the observer states, $F \in \mathbb{R}^{p \times n}$ and $L \in \mathbb{R}^{m \times q}$ are the feedback gain matrices, and $c > 0$ is a coupling gain. Here the algorithm (6) uses not only the relative outputs between each agent and its neighbors but also its own and neighbors' observer states. While relative outputs could be obtained through local measurements, the neighbors' observer states can only be obtained via communication. It can be shown that if the graph $\mathcal{G}$ has a directed spanning tree, consensus is reached using (6) for (4) if the matrices $A + BF$ and $A + c\lambda_i(\mathcal{L})LC$, where $\lambda_i(\mathcal{L}) \neq 0$, are Hurwitz. The observer-type consensus algorithm (6) can be seen as an extension of the single-system observer design to multi-agent systems. Here the *separation principle* of the traditional observer design still holds in the multi-agent setting in the sense that the feedback gain matrices F and L can be designed separately.

Consensus for Agents with Nonlinear Dynamics

In multi-agent applications, agents usually represent physical vehicles with special dynamics, especially nonlinear dynamics for the most part. Examples include Lagrangian systems for robotic manipulators and autonomous robots, nonholonomic systems for unicycles, attitude dynamics for rigid bodies, and general nonlinear systems. Similar to the consensus algorithms for linear multi-agent systems, the consensus algorithms used for these nonlinear agents are often designed based on state differences between each agent and its neighbors. But due to the inherent nonlinearity, the problem is more complicated and additional terms might be required in the algorithm design. The main techniques used in the consensus analysis for nonlinear multi-agent systems are often Lyapunov-based techniques (Lyapunov functions, passivity theory, nonlinear contraction analysis, and potential functions).

Early results on consensus for agents with nonlinear dynamics primarily focus on undirected graphs to exploit the symmetry to facilitate the construction of Lyapunov function candidates. Unfortunately, the extension from an undirected graph to a directed one is nontrivial.

For example, the directed graph does not preserve the passivity properties in general. Moreover, the directed graph could cause difficulties in the design of (positive-definite) Lyapunov functions. One approach is to integrate the nonnegative left eigenvector of the Laplacian matrix associated with the zero eigenvalue into the Lyapunov function, which is valid for strongly connected graphs and has been applied in some problems. Another approach is based on sliding mode control. The idea is to design a sliding surface for reaching a consensus. Taking multiple Lagrangian systems as an example, the agent dynamics are represented by

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) = \tau_i,$$
$$i = 1, \cdots, n, \tag{7}$$

where $q_i \in \mathbb{R}^p$ is the vector of generalized coordinates, $M_i(q_i) \in \mathbb{R}^{p \times p}$ is the symmetric positive-definite inertia matrix, $C_i(q_i, \dot{q}_i)\dot{q}_i \in \mathbb{R}^p$ is the vector of Coriolis and centrifugal torques, $g_i(q_i) \in \mathbb{R}^p$ is the vector of gravitational torque, and $\tau_i \in \mathbb{R}^p$ is the vector of control torque on the ith agent. The sliding surface can be designed as

$$s_i = \dot{q}_i - \dot{q}_{ri} = \dot{q}_i + \alpha \sum_{j \in \mathcal{N}_i} a_{ij}(q_i - q_j) \tag{8}$$

where α is a positive scalar. Note that when $s_i = 0$, (8) is actually the closed-loop system of a consensus algorithm for single integrators. Then if the control torque τ_i can be designed using only local information from neighbors to drive s_i to zero, consensus will be reached as s_i can be treated as a vanishing disturbance to a system that reaches consensus exponentially.

It is generally very challenging to deal with general directed or switching graphs for agents with more complicated dynamics other than single-integrator dynamics. In some cases, the challenge could be overcome by introducing and updating additional auxiliary variables (often observer-based algorithms) and exchanging these variables between neighbors (see, e.g., (6)). In the algorithm design, the agents might use not only relative physical states between

neighbors but also local auxiliary variables from neighbors. While relative physical states could be obtained through sensing, the exchange of auxiliary variables can only be achieved by communication. Hence such generalization is obtained at the price of increased communication between the neighboring agents. Unlike some other algorithms, it is generally impossible to implement the algorithm relying on purely relative sensing between neighbors without the need for communication.
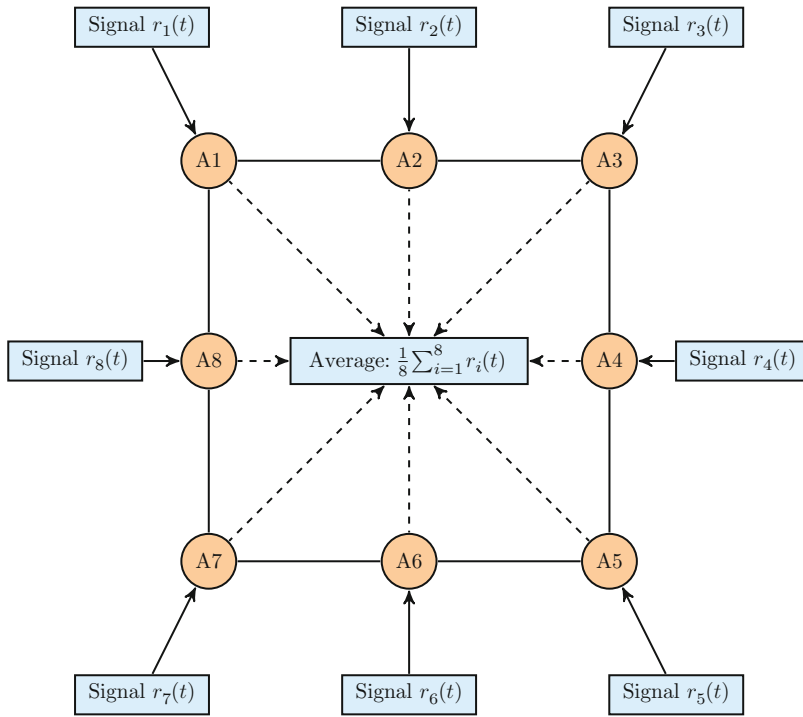
Averaging Algorithms

Existing distributed averaging algorithms are primarily static averaging algorithms based on linear local average iterations or gossip iterations. These algorithms are capable of computing the average of the initial conditions of all agents (or constant signals) in a network. In particular, the linear local-average-iteration algorithms are usually synchronous, where at each iteration each agent repeatedly updates its state to be the average of those of its neighbors. The gossip algorithms are asynchronous, where at each iteration a random pair of agents are selected to exchange their states and update them to be the average of the two. Dynamic averaging algorithms are of significance when there exist time-varying signals. The objective is to compute the average of these time-varying signals in a distributed manner.

Static Averaging
Take a linear local-average-iteration algorithm as an example. The results in this section follow from Tsitsiklis et al. (1986), Jadbabaie et al. (2003), and Olfati-Saber et al. (2007). Let x_i be the information state of agent i. A linear local-average-iteration-type algorithm has the form

$$x_i[k + 1] = \sum_{j \in \mathcal{N}_i[k]} a_{ij}[k]x_j[k], \quad i = 1, \ldots, n, \tag{9}$$

where k denotes a communication event, $\mathcal{N}_i[k]$ denotes the neighbor set of agent i, and $a_{ij}[k]$ is the (i, j) entry of the adjacency matrix $\mathcal{A}$ of the graph $\mathcal{G}$ that represents the communication

Averaging Algorithms and Consensus, Fig. 3 Illustration of distributed averaging of multiple (time-varying) signals. Here A_i denotes agent i and $r_i(t)$ denotes a (time-varying) signal associated with agent i. Each agent needs to compute the average of all agents' signals but can communicate with only its neighbors

topology at time k, with the additional assumption that $\mathcal{A}$ is row stochastic and $a_{ii}[k] > 0$ for all $i = 1, \ldots, n$. Intuitively, the information state of each agent is updated as the weighted average of its current state and the current states of its neighbors at each iteration. Note that an agent maintains its current state if it does not exchange information with other agents at that event instant. In fact, a discretized version of the closed-loop system of (1) using (2) (with a sufficiently small sampling period) takes in the form of (9). The objective here is for all agents to compute the average of their initial states by communicating with only their neighbors. That is, each $x_i[k]$ approaches $\frac{1}{n}\sum_{j=1}^{n} x_j[0]$ eventually. To compute the average of multiple constant signals c_i, we could simply set $x_i[0] = c_i$. The algorithm (9) can be written in matrix form as $x[k+1] = \mathcal{A}[k]x[k]$, where x is a column stack vector of all x_i and $\mathcal{A}[k] = [a_{ij}[k]]$ is a row-stochastic matrix.

When the graph $\mathcal{G}$ (and hence the matrix $\mathcal{A}$) is fixed, the convergence of the algorithm (9) can be analyzed by studying the eigenvalues and eigenvectors of the row-stochastic matrix $\mathcal{A}$. Because all diagonal entries of $\mathcal{A}$ are positive, Gershgorin's disc theorem implies that all eigenvalues of $\mathcal{A}$ are either within the open unit disk or at one. When the graph $\mathcal{G}$ is strongly connected, the Perron-Frobenius theorem implies that $\mathcal{A}$ has a simple eigenvalue at one with an associated right eigenvector $\mathbf{1}$ and an associated positive left eigenvector. Hence when $\mathcal{G}$ is strongly connected, it turns out that $\lim_{k\to\infty} \mathcal{A}^k = \mathbf{1}v^T$, where v^T is a positive left eigenvector of $\mathcal{A}$ associated with the eigenvalue one and satisfies $v^T\mathbf{1} = 1$. Note that $x[k] = \mathcal{A}^k x[0]$. Hence, each agent's state $x_i[k]$ approaches $v^T x[0]$ eventually. If it can be further ensured that $v = \frac{1}{n}\mathbf{1}$, then averaging is achieved. It can be shown that the agents' states converge to the average of their initial values if and only if the directed graph $\mathcal{G}$ is both strongly connected and balanced or the undirected graph $\mathcal{G}$ is connected. When the graph is switching, the convergence of the algorithm (9) can be analyzed by studying the

product of row-stochastic matrices. Such analysis is closely related to Markov chains. It can be shown that the agents' states converge to the average of their initial values if the directed graph $\mathcal{G}$ is balanced at each communication event and strongly connected in a joint manner or the undirected graph $\mathcal{G}$ is jointly connected.

Dynamic Averaging

In a more general setting, there exist n time-varying signals, $r_i(t)$, $i = 1, \ldots, n$, which could be an external signal or an output from a dynamical system. Here $r_i(t)$ is available to only agent i and each agent can exchange information with only its neighbors. Each agent maintains a local estimate, denoted by $x_i(t)$, of the average of all the signals $\bar{r}(t) \triangleq \frac{1}{n} \sum_{k=1}^{n} r_k(t)$. The objective is to design a distributed algorithm for agent i based on $r_i(t)$ and $x_j(t)$, $j \in \mathcal{N}_i(t)$, such that all agents will finally track the average that changes over time. That is, $\|x_i(t) - \bar{r}(t)\|$, $i = 1, \ldots, n$, approaches zero eventually. Such a dynamic averaging idea finds applications in distributed sensor fusion with time-varying measurements (Bai et al. 2011b; Spanos and Murray 2005) and distributed estimation and tracking (Yang et al. 2008).

Figure 3 illustrates the dynamic averaging idea. If there exists a central station that can always access the signals of all agents, then it is trivial to compute the average. Unfortunately, in a distributed context, where there does not exist a central station and each agent can only communicate with its local neighbors, it is challenging for each agent to compute the average that changes over time. While each agent could compute the average of its own and local neighbors' signals, this will not be the average of all signals.

When the signal $r_i(t)$ can be arbitrary but its derivative exists and is bounded almost everywhere, a distributed nonlinear nonsmooth algorithm is designed in Chen et al. (2012) as

$$\dot{\phi}_i(t) = \alpha \sum_{j \in \mathcal{N}_i} \operatorname{sgn}[x_j(t) - x_i(t)]$$

$$x_i(t) = \phi_i(t) + r_i(t), \quad i = 1, \ldots, n, \quad (10)$$

where α is a positive scalar, $\mathcal{N}_i$ denotes the neighbor set of agent i, $\operatorname{sgn}(\cdot)$ is the signum function defined componentwise, ϕ_i is the internal state of the estimator with $\phi_i(0) = 0$, and x_i is the estimate of the average $\bar{r}(t)$. Due to the existence of the discontinuous signum function, the solution of (10) is understood in the Filippov sense (Cortes 2008).

The idea behind the algorithm (10) is as follows. First, (10) is designed to ensure that $\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} r_i(t)$ holds for all time. Note that $\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} \phi_i(t) + \sum_{i=1}^{n} r_i(t)$. When the graph $\mathcal{G}$ is undirected and $\phi_i(0) = 0$, it follows that $\sum_{i=1}^{n} \phi_i(t) = \sum_{i=1}^{n} \phi_i(0) + \alpha \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \int_0^t \operatorname{sgn}[x_j(\tau) - x_i(\tau)] d\tau = 0$. As a result, $\sum_{i=1}^{n} x_i(t) = \sum_{i=1}^{n} r_i(t)$ holds for all time. Second, when $\mathcal{G}$ is connected, if the algorithm (10) guarantees that all estimates x_i approach the same value in *finite time*, then it can be guaranteed that each estimate approaches the average of all signals in finite time.

Summary and Future Research Directions

Averaging algorithms and consensus play an important role in distributed control of networked systems. While there is significant progress in this direction, there are still numerous open problems. For example, it is challenging to achieve averaging when the graph is not balanced. It is generally not clear how to deal with a general directed or switching graph for nonlinear agents or nonlinear algorithms when the algorithms are based on only interagent physical state coupling without the need for communicating additional auxiliary variables between neighbors. The study of consensus for multiple underactuated agents remains a challenge. Furthermore, when the agents' dynamics are heterogeneous, it is challenging to design consensus algorithms. In addition, in the existing study, it is often assumed that the agents are cooperative. When there exist faulty or malicious agents, the problem becomes more involved.

Cross-References

Bibliography

Agaev R, Chebotarev P (2000) The matrix of maximum out forests of a digraph and its applications. Autom Remote Control 61(9):1424–1450

Agaev R, Chebotarev P (2005) On the spectra of non-symmetric Laplacian matrices. Linear Algebra Appl 399:157–178

Bai H, Arcak M, Wen J (2011a) Cooperative control design: a systematic, passivity-based approach. Springer, New York

Bai H, Freeman RA, Lynch KM (2011b) Distributed Kalman filtering using the internal model average consensus estimator. In: Proceedings of the American control conference, San Francisco, pp 1500–1505

Bullo F, Cortes J, Martinez S (2009) Distributed control of robotic networks. Princeton University Press, Princeton

Chen F, Cao Y, Ren W (2012) Distributed average tracking of multiple time-varying reference signals with bounded derivatives. IEEE Trans Autom Control 57(12):3169–3174

Cortes J (2008) Discontinuous dynamical systems. IEEE Control Syst Mag 28(3):36–73

Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Autom Control 48(6):988–1001

Li Z, Duan Z, Chen G, Huang L (2010) Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint. IEEE Trans Circuits Syst I Regul Pap 57(1):213–224

Mesbahi M, Egerstedt M (2010) Graph theoretic methods for multiagent networks. Princeton University Press, Princeton

Moreau L (2005) Stability of multi-agent systems with time-dependent communication links. IEEE Trans Autom Control 50(2):169–182

Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

Qu Z (2009) Cooperative control of dynamical systems: applications to autonomous vehicles. Springer, London

Ren W, Beard RW (2008) Distributed consensus in multi-vehicle cooperative control. Springer, London

Ren W, Cao Y (2011) Distributed coordination of multi-agent networks. Springer, London

Spanos DP, Murray RM (2005) Distributed sensor fusion using dynamic consensus. In: Proceedings of the IFAC world congress, Prague

Tsitsiklis JN, Bertsekas DP, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE Trans Autom Control 31(9):803–812

Yang P, Freeman RA, Lynch KM (2008) Multi-agent coordination by decentralized estimation and control. IEEE Trans Autom Control 53(11):2480–2496

B

Backward Stochastic Differential Equations and Related Control Problems

Shige Peng
Shandong University, Jinan, Shandong Province, China

Synonyms

BSDE

Abstract

A conditional expectation of the form $Y_t = E[\xi + \int_t^T f_s ds | \mathcal{F}_t]$ is regarded as a simple and typical example of backward stochastic differential equation (abbreviated by BSDE). BSDEs are widely applied to formulate and solve problems related to stochastic optimal control, stochastic games, and stochastic valuation.

Keywords

Brownian motion; Feynman-Kac formula; Lipschitz condition; Optimal stopping

Definition

A typical real valued backward stochastic differential equation defined on a time interval $[0, T]$ and driven by a d-dim. Brownian motion B is

$$\begin{cases} dY_t = -f(t, Y_t, Z_t)dt + Z_t dB_t, \\ Y_T = \xi, \end{cases}$$

or its integral form

$$Y_t = \xi + \int_t^T f(s, \omega, Y_s, Z_s)ds - \int_t^T Z_s dB_s, \tag{1}$$

where ξ is a given random variable depending on the (canonical) Brownian path $B_t(\omega) = \omega(t)$ on $[0, T]$, $f(t, \omega, y, z)$ is a given function of the time t, the Brownian path ω on $[0, t]$, and the pair of variables $(y, z) \in \mathbb{R}^m \times \mathbb{R}^{m \times d}$. A solution of this BSDE is a pair of stochastic processes (Y_t, Z_t), the solution of the above equation, on $[0, T]$ satisfying the following constraint: for each t, the value of $Y_t(\omega)$, $Z_t(\omega)$ depends only on the Brownian path ω on $[0, t]$. Notice that, because of this constraint, the extra freedom Z_t is needed. For simplicity we set $d = m = 1$.

Often square-integrable conditions for ξ and f and Lipschitz condition for f with respect to (y, z) are assumed under which there exists a unique square-integrable solution (Y_t, Z_t) on $[0, T]$ (existence and uniqueness theorem of BSDE). We can also consider a multidimensional process Y and/or a multidimensional Brownian motion B, L^p-integrable conditions ($p \geq 1$) for ξ and f, as well as local Lipschitz conditions of f with respect to (y, z). If Y_t is real valued, we often call the equation a real valued BSDE.

We compare this BSDE with the classical stochastic differential equation (SDE):

$$dX_s = \sigma(X_s)dB_s + b(X_s)ds$$

with given initial condition $X_s|_{s=0} = x \in \mathbb{R}^n$. Its integral form is

$$X_t(\omega) = x + \int_0^t \sigma(X_s(\omega))dB_s(\omega)$$
$$+ \int_0^t b(X_s(\omega))ds. \qquad (2)$$

Linear backward stochastic differential equation was firstly introduced (Bismut 1973) in stochastic optimal control problems to solve the adjoint equation in the stochastic maximum principle of Pontryagin's type. The above existence and uniqueness theorem was obtained by Pardoux and Peng (1990). In the research domain of economics, this type of 1-dimensional BSDE was also independently derived by Duffie and Epstein (1992). Comparison theorem of BSDE was obtained in Peng (1992) and improved in El Karoui et al. (1997a). Nonlinear Feynman-Kac formula was obtained in Peng (1991, 1992) and improved in Pardoux and Peng (1992). BSDE is applied as a nonlinear Black-Scholes option pricing formula in finance. This formulation was given in El Karoui et al. (1997b). We refer to a recent survey in Peng (2010) for more details.

Hedging and Risk Measuring in Finance

Let us consider the following hedging problem in a financial market with a typical model of continuous time asset price: the basic securities consist of two assets, a riskless one called bond, and a risky security called stock. Their prices are governed by $dP_t^0 = P_t^0 r dt$, for the bond, and

$$dP_t = P_t[b dt + \sigma dB_t], \text{ for the stock.}$$

Here we only consider the situation where the volatility rate $\sigma > 0$. The case of

multidimensional stocks with degenerate volatility matrix σ can be treated by constrained BSDE. Assume that a small investor whose investment behavior cannot affect market prices and who invests at time $t \in [0, T]$ the amount π_t of his or her wealth Y_t in the security and π_t^0 in the bond, thus $Y_t = \pi_t^0 + \pi_t$. If his investment strategy is self-financing, then we have $dY_t = \pi_t^0 dP_t^0/P_t^0 + \pi_t dP_t/P_t$, thus

$$dY_t = (rY_t + \pi_t\sigma\theta)dt + \pi_t\sigma dB_t,$$

where $\theta = \sigma^{-1}(b - r)$. A strategy $(Y_t, \pi_t)_{t \in [0,T]}$ is said to be feasible if $Y_t \geq 0$, $t \in [0, T]$. A European path-dependent contingent claim settled at time T is a given nonnegative function of path $\xi = \xi((P_t)_{t \in [0,T]})$. A feasible strategy (Y, π) is called a hedging strategy against a contingent claim ξ at the maturity T if it satisfies

$$dY_t = (rY_t + \pi_t\sigma\theta)dt + \pi_t\sigma dB_t, \quad Y_T = \xi.$$

This problem can be regarded as finding a stochastic control π and an initial condition Y_0 such that the final state replicates the contingent claim ξ, i.e., $Y_T = \xi$. This type of replications is also called "exact controllability" in terms of stochastic control (see Peng 2005 for more general results).

Observe that $(Y, \pi\sigma)$ is the solution of the above BSDE. It is called a superhedging strategy if there exists an increasing process K_t, often called an accumulated consumption process, such that

$$dY_t = (rY_t + \pi_t\sigma\theta)dt + \pi_t\sigma dB_t - dK_t, \quad Y_T = \xi.$$

This type of strategies is often applied in a constrained market in which certain constraint $(Y_t, \pi_t) \in \Gamma$ is imposed. In fact a real market has many frictions and constraints. An example is the common case where interest rate R for borrowing money is higher than the bond rate r. The above equation for the hedging strategy becomes

$$dY_t = [rY_t + \pi_t\sigma\theta - (R - r)(\pi_t - Y_t)^+]$$
$$dt + \pi_t\sigma dB_t, Y_T = \xi,$$

where $[\alpha]^+ = \max\{\alpha, 0\}$. A short selling constraint $\pi_t \geq 0$ is also a typical requirement in markets. The method of constrained BSDE can be applied to this type of problems. BSDE theory provides powerful tools to the robust pricing and risk measures for contingent claims (see El Karoui et al. 1997a). For the dynamic risk measure under Brownian filtration, see Rosazza Gianin (2006), Peng (2004), Barrieu and El Karoui (2005), Hu et al. (2005), and Delbaen et al. (2010).

Comparison Theorem

The comparison theorem, for a real valued BSDE, tells us that, if (Y_t, Z_t) and $(\bar{Y}_t, \bar{Z}_t)$ are two solutions of BSDE (1) with terminal condition $Y_T = \xi$, $\bar{Y}_T = \bar{\xi}$ such that $\xi(\omega) \geq \bar{\xi}(\omega)$, $\omega \in \Omega$, then one has $Y_t \geq \bar{Y}_t$. This theorem holds if f and ξ, $\bar{\xi}$ satisfy the abovementioned L^2-integrability condition and f is a Lipschitz function in (y, z). This theorem plays the same important role as the maximum principle in PDE theory. The theorem also has several very interesting generalizations (see Buckdahn et al. 2000).

Stochastic Optimization and Two-Person Zero-Sum Stochastic Games

An important point of view is to regard an expectation value as a solution of a special type of BSDE. Consider an optimal control problem

$$\min_u J(u): \quad J(u) = E\left[\int_0^T l(X_s, u_s)ds + h(X_T)\right].$$

Here the state process X is controlled by the control process u_t which is valued in a control (compact) domain U through the following d-dimensional SDE

$$dX_s = b(X_s, u_s)ds + \sigma(X_s)dB_s$$

defined in a Wiener probability space $(\Omega, \mathcal{F}, P)$ with the Brownian motion $B_t(\omega) = \omega(t)$ which is the canonical process. Here we only discuss the case $\sigma \equiv I_d$ for simplicity. Observe that in fact the expected value $J(u)$ is $Y_0^u = E[Y_0^u]$, where Y_t^u solves the BSDE

$$Y_t^u = h(X_T) + \int_t^T l(X_s, u_s)ds - \int_t^T Z_s^u dB_s.$$

From Girsanov transformation, under the probability measure $\tilde{P}$ defined by

$$\frac{d\tilde{P}}{dP}|_T = \exp\left\{\int_0^T b(X_s, u_s)dB_s - \frac{1}{2}\int_0^T |b(X_s, u_s, v_s)|^2 ds\right\}$$

X_t is a Brownian motion, and the above BSDE is changed to

$$Y_t^u = h(X_T) + \int_t^T [l(X_s, u_s) + \langle Z_s^u, b(X_s, u_s)\rangle]ds - \int_t^T Z_s^u dX_s,$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product in $\mathbb{R}^d$. Notice that P and $\tilde{P}$ are absolutely continuous with each other. Compare this BSDE with the following one:

$$\hat{Y}_t = h(X_T) + \int_t^T H(X_s, \hat{Z}_s)ds - \int_t^T \hat{Z}_s dX_s, \tag{3}$$

where $H(x, z) := \inf_{u \in U} \{l(x, u) + \langle z, b(x, u)\rangle\}$. It is a direct consequence of the comparison theorem of BSDE that $\hat{Y}_0 \leq Y_0^u = J(u)$, for any admissible control u_t. Moreover, one can find a feedback control $\hat{u}$ such that $\hat{Y}_0 = J(\hat{u})$.

The above BSDE method has been introduced to solve the following two-person zero-sum game (Hamadène and Lepeltier 1995):

$$\max_v \min_u J(u, v), \quad J(u, v)$$
$$= E\left[\int_0^T l(X_s, u_s, v_s)ds + h(X_T)\right]$$

with

$$dX_s = b(X_s, u_s, v_s)ds + dB_s,$$

where (u_s, v_s) is formulated as above with compact control domains $u_s \in U$ and $v_s \in V$. In this case the equilibrium of the game exists if the following Isaac condition is satisfied:

$$H(x,z) := \max_{v \in V} \inf_{u \in U} \{l(x,u,v) + \langle z, b(x,u,v) \rangle\}$$
$$= \inf_{u \in U} \max_{v \in V} \{l(x,u,v) + \langle z, b(x,u,v) \rangle\},$$

and the equilibrium is also obtained through a BSDE (3) defined above.

Nonlinear Feynman-Kac Formula

A very interesting situation is when $f = g(X_t, y, z)$ and $Y_T = \varphi(X_T)$ in BSDE (1). In this case we have the following relation, called "nonlinear Feynman-Kac formula,"

$$Y_t = u(t, X_t), \quad Z_t = \sigma^T(X_t) \nabla u(t, X_t)$$

where $u = u(t, x)$ is the solution of the following quasilinear parabolic PDE:

$$\partial_t u + \mathcal{L} u + g(x, u, \sigma^T \nabla u) = 0, \qquad (4)$$

$$u(x, T) = \varphi(x), \qquad (5)$$

where $\mathcal{L}$ is the following, possibly degenerate, elliptic operator:

$$\mathcal{L}\varphi(x) = \frac{1}{2} \sum_{i,j=1}^{d} a_{ij}(x) \partial^2_{x_i x_j} \varphi(x)$$

$$+ \sum_{i=1}^{d} b_i(x) \partial_{x_i} \varphi(x), \quad a(x) = \sigma(x) \sigma^T(x).$$

Nonlinear Feynman-Kac formula can be used to solve a nonlinear PDE of form (4) to (5) by a BSDE (1) coupled with an SDE (2).

A general principle is, once we solve a BSDE driven by a Markov process X for which the terminal condition Y_T at time T depends only on X_T and the generator $f(t, \omega, y, z)$ also depends on the state X_t at each time t, then the corresponding solution of the BSDE is also state dependent, namely, $Y_t = u(t, X_t)$, where u is the solution

of the corresponding quasilinear PDE. Once Y_T and g are path functions of X, then the solution of the BSDE becomes also path dependent. In this sense, we can say that the PDE is in fact a "state-dependent BSDE,"and BSDE gives us a new generalization of "path-dependent PDE" of parabolic and/or elliptic types. This principle was illustrated in Peng (2010) for both quasilinear and fully nonlinear situations.

Observe that BSDE (1) and forward SDE (2) are only partially coupled. A fully coupled system of SDE and BSDE is called a forward-backward stochastic differential equation (FBSDE). It has the following form:

$$dX_t = b(t, X_t, Y_t, Z_t) dt + \sigma(t, X_t, Y_t, Z_t) dB_t,$$
$$X_0 = x \in \mathbb{R}^n,$$
$$-dY_t = f(t, X_t, Y_t, Z_t) dt - Z_t dB_t, \quad Y_T = \varphi(X_T).$$

In general the Lipschitz assumptions for b, σ, f, and φ w. r. t. (x, y, z) are not enough. Then Ma et al. (1994) have proposed a four-step scheme method of FBSDE for the nondegenerate Markovian case with σ independent of Z. For the case $\dim(x) = \dim(y) = n$, Hu and Peng (1995) proposed a new type of monotonicity condition. This method does not need to assume the coefficients to be deterministic. Peng and Wu (1999) have weakened the monotonicity condition. Observe that in the case where $b = \nabla_y H(x, y, z)$, $\sigma = \nabla_z H(x, y, z)$, and $f = \nabla_x H(x, y, z)$, for a given real valued function H convex in x concave in (y, z), the above FBSDE is called the stochastic Hamilton equation associated to a stochastic optimal control problem. We also refer to the book of Ma and Yong (1999) for a systematic exposition on this subject. For time-symmetric forward-backward stochastic differential equations and its relation with stochastic optimality, see Peng and Shi (2003) and Han et al. (2010).

Reflected BSDE and Optimal Stopping

If (Y, Z) solves the BSDE

$$dY_s = -g(s, Y_s, Z_s) ds + Z_s dB_s - dK_s, \quad Y_T = \xi,$$
$$(6)$$

where K is a càdlàg and increasing process with $K_0 = 0$ and $K_t \in L_P^2(\mathcal{F}_t)$, then Y or (Y, Z, K) is called a *supersolution* of the BSDE, or *g-supersolution*. This notion is often used for constrained BSDEs. A typical situation is as follows: for a given continuous adapted process $(L_t)_{t \in [0,T]}$, find a smallest g-supersolution (Y, Z, K) such that $Y_t \geq L_t$. This problem was initialed in El Karoui et al. (1997b). It is proved that this problem is equivalent to finding a triple (Y, Z, K) satisfying (4) and the following reflecting condition of Skorohod type:

$$Y_s \geq L_s, \quad \int_0^T (Y_s - L_s) dK_s = 0. \quad (7)$$

In fact $\tau^* := \inf\{t \in [0, T] : K_t > 0\}$ is the optimal stopping time associated to this BSDE. A well-known example is the pricing of American option.

Moreover, a new type of nonlinear Feynman-Kac formula was introduced: if all coefficients are given as in the formulation of the above nonlinear Feynman-Kac formula and $L_s = \Phi(X_s)$ where Φ satisfies the same condition as φ, then we have $Y_s = u(s, X_s)$, where $u = u(t, x)$ is the solution of the following variational inequality:

$$\min\{\partial_t u + \mathcal{L}u + g(x, u, \sigma^* Du), u - \Phi\}$$
$$= 0, \quad (t, x) \in [0, T] \times \mathbb{R}^n, \quad (8)$$

with terminal condition $u|_{t=T} = \varphi$. They also demonstrated that this reflected BSDE is a powerful tool to deal with contingent claims of American types in a financial market with constraints.

BSDE reflected within two barriers, a lower one L and an upper one U, was first investigated by Cvitanic and Karatzas (1996) where a type of nonlinear Dynkin games was formulated for a two-player model with zero-sum utility and each player chooses his own optimal exit time.

Stochastic optimal switching problems can be also solved by new types of oblique-reflected BSDEs.

A more general case of constrained BSDE is to find the smallest g-supersolution (Y, Z, K) with constraint $(Y_t, Z_t) \in \Gamma_t$ where, for each $t \in$ $[0, T]$, Γ_t (El Karoui and Quenez 1995; Cvitanic and Karatzas 1993; El Karoui et al. 1997a) for the problem of superhedging in a market with convex constrained portfolios (Cvitanic et al. 1998). The case with an arbitrary closed constraint was proved in Peng (1999).

Backward Stochastic Semigroup and g-Expectations

Let $\mathcal{E}_{t,T}^g[\xi] = Y_t$ where Y is the solution of BSDE (1). $(\mathcal{E}_{t,T}^g[\cdot])_{0 \leq t \leq T < \infty}$ has the (backward) semigroup property (Peng 1997)

$$\mathcal{E}_{s,t}^g[\mathcal{E}_{t,T}^g[\xi]] = \mathcal{E}_{s,T}^g[\xi], \quad \mathcal{E}_{T,T}^g[\xi]$$
$$= \xi, \quad 0 \leq s \leq t \leq T.$$

For a real valued BSDE, by the comparison theorem, the semigroup is monotone: $\mathcal{E}_{t,T}^g[\xi] \geq \mathcal{E}_{t,T}^g[\bar{\xi}]$, if $\xi \geq \bar{\xi}$. If moreover $g|_{z=0} = 0$, then the semigroup is constant preserving: $\mathcal{E}_{t,T}^g[c] \equiv c$. Thus the semigroup forms in fact a nonlinear expectation called g-expectation (since this nonlinear expectation is totally determined by the generator g).

This notion allows us to establish a nonlinear g-martingale theory, e.g., g-supermartingale decomposition theorem. Peng (1999) claims that, if Y is a square-integrable càdlàg g-supermartingale, then it has the unique decomposition: there exists a unique predictable, increasing, and càdlàg process A such that Y solves

$$-dY_t = g(t, Y_t, Z_t)dt + dA_t - Z_t dB_t.$$

A theoretically challenging and practically important problem is as follows: given an abstract family of expectations $(\mathcal{E}_{s,t}[\cdot])_{s \leq t}$ satisfying the same backward semigroup properties as these of g-expectation, can we find a function g such that $\mathcal{E}_{s,t} \equiv \mathcal{E}_{s,t}^g$? Coquet, Hu et al. (2005) proved that if $\mathcal{E}$ is dominated by g_μ-expectation with $g_\mu(z) = \mu|z|$ for a large enough constant $\mu > 0$, then there exists a unique function $g = g(t, \omega, z)$ satisfying μ-Lipschitz condition such that $(\mathcal{E}_{s,t}[\cdot])_{s \leq t}$ is in fact a g-expectation. For a concave dynamic expectation with an assumption much weaker than

the above domination condition, we can still find a function $g = g(t, z)$ with possibly singular values (Delbaen et al. 2010). For the case without the assumption of constant preservation, see Peng (2005). In practice, the above criterion is very useful to test whether a dynamic pricing mechanism of contingent contracts can be represented through a concrete function g.

A serious challenging problem in the stochastic control theory is as follows: it is based on a given probability space $(\Omega, \mathcal{F}, P)$. But in most practical situations, it is far from being true. In many risky situations, it is necessary to consider the uncertainty of the probability measures themselves, e.g., $\{P_\theta\}_{\theta \in \Theta}$, namely, the well-known Knightian uncertainty (Knight 1921). A new framework of G-expectation space $(\Omega, \mathcal{H}, \hat{\mathbb{E}})$ and the corresponding random and stochastic analysis (Itô's analysis) is introduced (see Peng 2007, 2010 and Soner et al. 2012) to replace the probability framework $(\Omega, \mathcal{F}, P)$. g-expectation is a special and typical case in this new theory.

Cross-References

▶ Numerical Methods for Continuous-Time Stochastic Control Problems
▶ Risk-Sensitive Stochastic Control
▶ Stochastic Dynamic Programming
▶ Stochastic Linear-Quadratic Control
▶ Stochastic Maximum Principle

Recommended Reading

BSDE theory applied in maximization of stochastic control can be found in the book of Yong and Zhou (1999); stochastic control problem in finance in El Karoui et al. (1997a); optimal stopping and reflected BSDE in El Karoui et al. (1997b); Maximization under Knightian uncertainty using nonlinear expectation can be found in Chen and Epstein (2002) and a survey paper in Peng (2010).

Bibliography

Barrieu P, El Karoui N (2005) Inf-convolution of risk measures and optimal risk transfer. Financ Stoch 9: 269–298

Bismut JM (1973) Conjugate convex functions in optimal stochastic control. J Math Anal Apl 44: 384–404

Buckdahn R, Quincampoix M, Rascanu A (2000) Viability property for a backward stochastic differential equation and applications to partial differential equations. Probab Theory Relat Fields 116(4): 485–504

Chen Z, Epstein L (2002) Ambiguity, risk and asset returns in continuous time. Econometrica 70(4): 1403–1443

Coquet F, Hu Y, Memin J, Peng S (2002) Filtration consistent nonlinear expectations and related g-Expectations. Probab Theory Relat Fields 123: 1–27

Cvitanic J, Karatzas I (1993) Hedging contingent claims with constrained portfolios. Ann Probab 3(4):652–681

Cvitanic J, Karatzas I (1996) Backward stochastic differential equations with reflection and Dynkin games. Ann Probab 24(4):2024–2056

Cvitanic J, Karatzas I, Soner M (1998) Backward stochastic differential equations with constraints on the gains-process. Ann Probab 26(4):1522–1551

Delbaen F, Rosazza Gianin E, Peng S (2010) Representation of the penalty term of dynamic concave utilities. Finance Stoch 14:449–472

Duffie D, Epstein L (1992) Appendix C with costis skiadas, stochastic differential utility. Econometrica 60(2):353–394

El Karoui N, Quenez M-C (1995) Dynamic programming and pricing of contingent claims in an incomplete market. SIAM Control Optim 33(1): 29–66

El Karoui N, Peng S, Quenez M-C (1997a) Backward stochastic differential equation in finance. Math Financ 7(1):1–71

El Karoui N, Kapoudjian C, Pardoux E, Peng S, Quenez M-C (1997b) Reflected solutions of backward SDE and related obstacle problems for PDEs. Ann Probab 25(2):702–737

Hamadène S, Lepeltier JP (1995) Zero-sum stochastic differential games and backward equations. Syst Control Lett 24(4):259–263

Han Y, Peng S, Wu Z (2010) Maximum principle for backward doubly stochastic control systems with applications. SIAM J Control 48(7):4224–4241

Hu Y, Peng S (1995) Solution of forward-backward stochastic differential-equations. Probab Theory Relat Fields 103(2):273–283

Hu Y, Imkeller P, Müller M (2005) Utility maximization in incomplete markets. Ann Appl Probab 15(3): 1691–1712

Knight F (1921) Risk, uncertainty and profit. Hougton Mifflin Company, Boston. (Dover, 2006)

Ma J, Yong J (1999) Forward-backward stochastic differential equations and their applications. Lecture notes in mathematics, vol 1702. Springer, Berlin/New York

Ma J, Protter P, Yong J (1994) Solving forward–backward stochastic differential equations explicitly, a four step scheme. Probab Theory Relat Fields 98: 339–359

Pardoux E, Peng S (1990) Adapted solution of a backward stochastic differential equation. Syst Control Lett 14(1):55–61

Pardoux E, Peng S (1992) Backward stochastic differential equations and quasilinear parabolic partial differential equations, Stochastic partial differential equations and their applications. In: Proceedings of the IFIP. Lecture notes in CIS, vol 176. Springer, pp 200–217

Peng S (1991) Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. Stochastics 37:61–74

Peng S (1992) A generalized dynamic programming principle and hamilton-jacobi-bellmen equation. Stochastics 38:119–134

Peng S (1994) Backward stochastic differential equation and exact controllability of stochastic control systems. Prog Nat Sci 4(3):274–284

Peng S (1997) BSDE and stochastic optimizations. In: Yan J, Peng S, Fang S, Wu LM (eds) Topics in stochastic analysis. Lecture notes of xiangfan summer school, chap 2. Science Publication (in Chinese, 1995)

Peng S (1999) Monotonic limit theorem of BSDE and nonlinear decomposition theorem of Doob-Meyer's type. Probab Theory Relat Fields 113(4):473–499

Peng S (2004) Nonlinear expectation, nonlinear evaluations and risk measurs. In: Back K, Bielecki TR, Hipp C, Peng S, Schachermayer W (eds) Stochastic methods in finance lectures, C.I.M.E.-E.M.S. Summer School held in Bressanone/Brixen, LNM vol 1856. Springer, pp 143–217. (Edit. M. Frittelli and W. Runggaldier)

Peng S (2005) Dynamically consistent nonlinear evaluations and expectations. arXiv:math. PR/ 0501415 v1

Peng S (2007) G-expectation, G-Brownian motion and related stochastic calculus of Itô's type. In: Benth et al. (eds) Stochastic analysis and applications, The Abel Symposium 2005, Abel Symposia, pp 541–567. Springer

Peng S (2010) Backward stochastic differential equation, nonlinear expectation and their applications. In: Proceedings of the international congress of mathematicians, Hyderabad

Peng S, Shi Y (2003) A type of time-symmetric forward-backward stochastic differential equations. C R Math Acad Sci Paris 336:773–778

Peng S, Wu Z (1999) Fully coupled forward-backward stochastic differential equations and applications to optimal control. SIAM J Control Optim 37(3): 825–843

Rosazza Gianin E (2006) Risk measures via G-expectations. Insur Math Econ 39:19–34

Soner M, Touzi N, Zhang J (2012) Wellposedness of second order backward SDEs. Probab Theory Relat Fields 153(1–2):149–190

Yong J, Zhou X (1999) Stochastic control. Applications of mathematics, vol 43. Springer, New York

Basic Numerical Methods and Software for Computer Aided Control Systems Design

Volker Mehrmann[1] and Paul Van Dooren[2]
[1]Institut für Mathematik MA 4-5, Technische Universität Berlin, Berlin, Germany
[2]ICTEAM: Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium

Abstract

Basic principles for the development of computational methods for the analysis and design of linear time-invariant systems are discussed. These have been used in the design of the subroutine library SLICOT. The principles are illustrated on the basis of a method to check the controllability of a linear system.

Keywords

Accuracy; Basic numerical methods; Benchmarking; Controllability; Documentation and implementation standards; Efficiency; Software design

Introduction

Basic numerical methods for the analysis and design of dynamical systems are at the heart of most techniques in systems and control theory that are used to describe, control, or optimize industrial and economical processes. There are many methods available for all the different tasks in systems and control, but even though most of these methods are based on sound theoretical principles, many of them still fail when applied to real-life problems. The reasons for this may be quite diverse, such as the fact that the system dimensions are very large, that the underlying problem is very sensitive to small changes in the data, or that the method lacks numerical

robustness when implemented in a finite precision environment.

To overcome such failures, major efforts have been made in the last few decades to develop robust, well-implemented, and standardized software packages for computer-aided control systems design (Grübel 1983; Nag Slicot 1990; Wieslander 1977). Following the standards of modern software design, such packages should consist of numerically robust routines with known performance in terms of reliability and efficiency that can be used to form the basis of more complex control methods. Also to avoid duplication and to achieve efficiency and portability to different computational environments, it is essential to make maximal use of the established standard packages that are available for numerical computations, e.g., the **B**asic **L**inear **A**lgebra **S**ubroutines (BLAS) (Dongarra et al. 1990) or the **L**inear **A**lgebra **Pack**ages (LAPACK) (Anderson et al. 1992). On the basis of such standard packages, the next layer of more complex control methods can then be built in a robust way.

In the late 1980s, a working group was created in Europe to coordinate efforts and integrate and extend the earlier software developments in systems and control. Thanks to the support of the European Union, this eventually led to the development of the **S**ubroutine **L**ibrary **in** **Co**ntrol **T**heory (SLICOT) (Benner et al. 1999; SLICOT 2012). This library contains most of the basic computational methods for control systems design of linear time-invariant control systems.

An important feature of this and similar kind of subroutine libraries is that the development of further higher level methods is not restricted by specific requirements of the languages or data structures used and that the routines can be easily incorporated within other more user-friendly software systems (Gomez et al. 1997; MATLAB 2013). Usually, this *low-level reusability* can only be achieved by using a general-purpose programming language like C or Fortran.

We cannot present all the features of the SLICOT library here. Instead, we discuss its general philosophy in section "The Control Subroutine Library SLICOT" and illustrate these concepts in section "An Illustration" using one specific task, namely, checking the controllability of a system. We refer to SLICOT (2012) for more details on SLICOT and to Varga (2004) for a general discussion on numerical software for systems and control.

The Control Subroutine Library SLICOT

When designing a subroutine library of basic algorithms, one should make sure that it satisfies certain basic requirements and that it follows a strict standardization in implementation and documentation. It should also contain standardized test sets that can be used for benchmarking, and it should provide means for maintenance and portability to new computing environments. The subroutine library SLICOT was designed to satisfy the following basic recommendations that are typically expected in this context (Benner et al. 1999).

Robustness: A subroutine must either return reliable results or it must return an error or warning indicator, if the problem has not been well posed or if the problem does not fall in the class to which the algorithm is applicable or if the problem is too ill-conditioned to be solved in a particular computing environment.

Numerical stability and accuracy: Subroutines are supposed to return results that are as good as can be expected when working at a given precision. They also should provide an option to return a parameter estimating the accuracy actually achieved.

Efficiency: An algorithm should never be chosen for its speed if it fails to meet the usual standards of robustness, numerical stability, and accuracy, as described above. Efficiency must be evaluated, e.g., in terms of the number of floating-point operations, the memory requirements, or the number and cost of iterations to be performed.

Modern computer architectures: The requirements of modern computer architectures must be taken into account, such as shared

or distributed memory parallel processors, which are the standard environments of today. The differences in the various architectures may imply different choices of algorithms.

Comprehensive functional coverage: The routines of the library should solve control systems relevant computational problems and try to cover a comprehensive set of routines to make it functional for a wide range of users. The SLICOT library covers most of the numerical linear algebra methods needed in systems analysis and synthesis problems for standard and generalized state space models, such as Lyapunov, Sylvester, and Riccati equation solvers, transfer matrix factorizations, similarity and equivalence transformations, structure exploiting algorithms, and condition number estimators.

The implementation of subroutines for a library should be highly standardized, and it should be accompanied by a well-written online documentation as well as a user manual (see, e.g., standard Denham and Benson 1981; Working Group Software 1996) which is compatible with that of the LAPACK library (Anderson et al. 1992). Although such highly restricted standards often put a heavy burden on the programmer, it has been observed that it has a high importance for the reusability of software and it also has turned out to be a very valuable tool in teaching students how to implement algorithms in the context of their studies.

Benchmarking

In the validation of numerical software, it is extremely important to be able to test the correctness of the implementation as well as the performance of the method, which is one of the major steps in the construction of a software library. To achieve this, one needs a standardized set of benchmark examples that allows an evaluation of a method with respect to correctness, accuracy, and efficiency and to analyze the behavior of the method in extreme situations, i.e., on problems where the limit of the possible accuracy is reached. In the context of basic systems and control methods, several such benchmark collections have been developed (see, e.g., Benner et al. 1997; Frederick 1998, or http://www.slicot.org/index.php?site=benchmarks).

Maintenance, Open Access, and Archives

It is a major challenge to maintain a well-developed library accessible and usable over time when computer architectures and operating systems are changing rapidly, while keeping the library open for access to the user community. This usually requires financial resources that either have to be provided by public funding or by licensing the commercial use.

In the SLICOT library, this challenge has been addressed by the formation of the Niconet Association (http://www.niconet-ev.info/en/) which provides the current versions of the codes and all the documentations. Those of Release 4.5 are available under the GNU General Public License or from the archives of http://www.slicot.org/.

An Illustration

To give an illustration for the development of a basic control system routine, we consider the specific problem of checking controllability of a linear time-invariant control system. A linear time-invariant control problem has the form

$$\frac{dx}{dt} = Ax + Bu, \ t \in [t_0, \infty) \qquad (1)$$

Here x denotes the *state* and u the *input function*, and the system matrices are typically of the form $A \in \mathbb{R}^{n,n}, \ B \in \mathbb{R}^{n,m}$.

One of the most important topics in control is the question whether by an appropriate choice of input function $u(t)$ we can control the system from an arbitrary state to the null state. This property, called *controllability*, can be characterized by one of the following equivalent conditions (see Paige 1981).

Theorem 1 *The following are equivalent:*

(i) System (1) is controllable.

(ii) Rank $[B, AB, A^2B, \cdots, A^{n-1}B] = n$.

(iii) Rank $[B, A - \lambda I] = n \quad \forall \lambda \in \mathbb{C}$.

(iv) $\exists F$ *such that* A *and* $A + BF$ *have no common eigenvalues.*

The conditions of Theorem 1 are nice for theoretical purposes, but none of them is really adequate for the implementation of an algorithm that satisfies the requirements described in the previous section. Condition (ii) creates difficulties because the controllability matrix $K = [B, AB, A^2B, \cdots, A^{n-1}B]$ will be highly corrupted by roundoff errors. Condition (iii) can simply not be checked in finite time. However, it is sufficient to check this condition only for the eigenvalues of A, but this is extremely expensive. And finally, condition (iv) will almost always give disjoint spectra between A and $A + BF$ since the computation of eigenvalues is sensitive to roundoff.

To devise numerical procedures, one often resorts to the computation of canonical or condensed forms of the underlying system. To obtain such a form one employs controllability preserving linear transformations $x \mapsto Px$, $u \mapsto Qu$ with nonsingular matrices $P \in \mathbb{R}^{n,n}$, $Q \in \mathbb{R}^{m,m}$. The canonical form under these transformations is the Luenberger form (see Luenberger 1967). This form allows to check the controllability using the above criterion (iii) by simple inspection of the condensed matrices. This is ideal from a theoretical point of view but is very sensitive to small perturbations in the data, in particular because the transformation matrices may have arbitrary large norm, which may lead to large errors.

For the implementation as robust numerical software one uses instead transformations with real orthogonal matrices P, Q that can be implemented in a backward stable manner, i.e., the resulting backward error is bounded by a small constant times the unit roundoff **u** of the finite precision arithmetic, and employs for reliable rank determinations the well-known singular value decomposition (SVD) (see, e.g., Golub and Van Loan 1996).

Theorem 2 (Singular value decomposition) *Given* $A \in \mathbb{R}^{n,m}$, *then there exist orthogonal matrices* U, V *with* $U \in \mathbb{R}^{n,n}$, $V \in \mathbb{R}^{m,m}$, *such that* $A = U\Sigma V^T$ *and* $\Sigma \in \mathbb{R}^{n,m}$ *is quasi-diagonal, i.e.,*

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \text{ where } \Sigma_r = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix},$$

and the nonzero singular values σ_i *are ordered as* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$.

The SVD presents the *best* way to determine (numerical) ranks of matrices in finite precision arithmetic by counting the number of singular values satisfying $\sigma_j \geq \mathbf{u}\sigma_1$ and by putting those for which $\sigma_j < \mathbf{u}\sigma_1$ equal to zero. The computational method for the SVD is well established and analyzed, and it has been implemented in the LAPACK routine SGESVD (see http://www.netlib.org/lapack/). A faster but less reliable alternative to compute the numerical rank of a matrix A is its QR factorization with pivoting (see, e.g., Golub and Van Loan 1996).

Theorem 3 (QRE decomposition) *Given* $A \in \mathbb{R}^{n,m}$, *then there exists an orthogonal matrix* $Q \in \mathbb{R}^{n,n}$ *and a permutation* $E \in \mathbb{R}^{m,m}$, *such that* $A = QRE^T$ *and* $R \in \mathbb{R}^{n,m}$ *is trapezoidal, i.e.,*

$$R = \begin{bmatrix} r_{11} & \dots & r_{1l} & \dots & r_{1m} \\ & \ddots & & & \vdots \\ & & r_{ll} & \dots & r_{lm} \\ 0 & & & 0 & \end{bmatrix}.$$

and the nonzero diagonal entries r_{ii} *are ordered as* $r_{11} \geq \cdots \geq r_{ll} > 0$.

The (numerical) rank in this case is again obtained by counting the diagonal elements $r_{ii} \geq \mathbf{u}r_{11}$.

One can use such orthogonal transformations to construct the controllability staircase form (see Van Dooren 1981).

Theorem 4 (Staircase form) *Given matrices* $A \in R^{n,n}$, $B \in R^{n,m}$, *then there exist orthogonal matrices* P, Q *with* $P \in \mathbb{R}^{n,n}$, $Q \in \mathbb{R}^{m,m}$, *so that*

$$PAP^T = \begin{bmatrix} A_{11} & \cdots & & \cdots & A_{1,r-1} & A_{1,r} \\ A_{21} & \ddots & & & \vdots & \vdots \\ & \ddots & \ddots & & \vdots & \vdots \\ & & A_{r-1,r-2} & A_{r-1,r-1} & A_{r-1,r} \\ 0 & \cdots & 0 & 0 & A_{rr} \end{bmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ n_{r-1} \\ n_r \end{matrix}$$
$$ \begin{matrix} n_1 & \cdots & n_{r-2} & n_{r-1} & n_r \end{matrix}$$

$$PBQ = \begin{bmatrix} B_1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{matrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ \vdots \\ n_r \end{matrix} \qquad (2)$$
$$ \begin{matrix} n_1 & m - n_1 \end{matrix}$$

where $n_1 \geq n_2 \geq \cdots \geq n_{r-1} \geq n_r \geq 0, n_{r-1} > 0$, $A_{i,i-1} = \begin{bmatrix} \Sigma_{i,i-1} & 0 \end{bmatrix}$, with nonsingular blocks $\Sigma_{i,i-1} \in \mathbb{R}^{n_i,n_i}$ and $B_1 \in \mathbb{R}^{n_1,n_1}$.

Notice that when using the reduced pair in condition (iii) of Theorem 1, the controllability condition is just $n_r = 0$, which is simply checked by inspection. A numerically stable algorithm to compute the staircase form of Theorem 4 is given below. It is based on the use of the singular value decomposition, but one could also have used instead the QR decomposition with column pivoting.

Staircase Algorithm
Input: $A \in \mathbb{R}^{n,n}, B \in \mathbb{R}^{n,m}$
Output: PAP^T, PBQ in the form (2), P, Q orthogonal

Step 0: Perform an SVD $B = U_B \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix} V_B^T$ with nonsingular and diagonal $\Sigma_B \in \mathbb{R}^{n_1,n_1}$. Set $P := U_B^T, Q := V_B$, so that

$$A := U_B^T A U_B = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

$$B := U_B^T B V_B = \begin{bmatrix} \Sigma_B & 0 \\ 0 & 0 \end{bmatrix}$$

with A_{11} of size $n_1 \times n_1$.

Step 1: Perform an SVD $A_{21} = U_{21} \begin{bmatrix} \Sigma_{21} & 0 \\ 0 & 0 \end{bmatrix} V_{21}^T$ with nonsingular and diagonal $\Sigma_{21} \in \mathbb{R}^{n_2,n_2}$. Set

$$P_2 := \begin{bmatrix} V_{21}^T & 0 \\ 0 & U_{21}^T \end{bmatrix}, \quad P := P_2 P$$

so that

$$A := P_2 A P_2^T =: \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & A_{32} & A_{33} \end{bmatrix},$$

$$B := P_2 B =: \begin{bmatrix} B_1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where $A_{21} = \begin{bmatrix} \Sigma_{21} & 0 \end{bmatrix}$, and $B_1 := V_{21}^T \Sigma_B$.

Step 2:
i = 3
DO WHILE ($n_{i-1} > 0$ AND $A_{i,i-1} \neq 0$).
 Perform an SVD of $A_{i,i-1} = U_{i,i-1} \begin{bmatrix} \Sigma_{i,i-1} & 0 \\ 0 & 0 \end{bmatrix} V_{i,i-1}^T$ with
 $\Sigma_{i,i-1} \in \mathbb{R}^{n_i,n_i}$ nonsingular and diagonal.
 Set

$$P_i := \begin{bmatrix} I_{n_1} \\ & \ddots \\ & & I_{n_{i-2}} \\ & & & V_{i,i-1}^T \\ & & & & U_{i,i-1}^T \end{bmatrix}, P := P_i P,$$

so that

$$A := P_i A P_i^T =: \begin{bmatrix} A_{11} & \cdots & & & A_{1,i+1} \\ A_{21} & \ddots & & & A_{2,i+1} \\ & \ddots & \ddots & & \vdots \\ & & A_{i,i-1} & A_{i,i} & \vdots \\ 0 & & & A_{i+1,i} & A_{i+1,i+1} \end{bmatrix}$$

where $A_{i,i-1} = \begin{bmatrix} \Sigma_{i,i-1} & 0 \end{bmatrix}$.
$i := i + 1$
END
$r := i$

It is clear that this algorithm will stop with $n_i = 0$ or $A_{i,i-1} = 0$. In every step, the remaining block shrinks at least by 1 row/column, as long as Rank $A_{i,i-1} > 1$, so that the algorithm stops after maximally $n - 1$ steps. It has been shown in Van Dooren (1981) that system (1) is controllable if and only if in the staircase form of (A, B) one has $n_r = 0$.

It should be noted that the updating transformations P_i of this algorithm will affect previously created "stairs" so that the blocks denoted as $\Sigma_{i,i-1}$ will not be diagonal anymore, but their singular values are unchanged. This is critical in the decision about the controllability of the pair (A, B) since it depends on the numerical rank of the submatrices $A_{i,i-1}$ and B (see Demmel and Kågström 1993). Based on this and a detailed error and perturbation analysis, the Staircase Algorithm has been implemented in the SLICOT routine `AB01OD`, and it uses in the worst-case $\mathcal{O}(n^4)$ flops (a "flop" is an elementary floating-point operation $+, -, *,$ or $/$). For efficiency reasons, the SLICOT routine `AB01OD` does not use SVDs for rank decisions, but QR decompositions with column pivoting. When applying the corresponding orthogonal transformations to the system without accumulating them, the complexity can be reduced to $\mathcal{O}(n^3)$ flops. It has been provided with error bounds, condition estimates, and warning strategies.

Summary and Future Directions

We have presented the SLICOT library and the basic principles for the design of such basic subroutine libraries. To illustrate these principles, we have presented the development of a method for checking controllability for a linear time-invariant control system. But the SLICOT library contains much more than that. It essentially covers most of the problems listed in the selected reprint volume (Patel et al. 1994). This volume contained in 1994 the state of the art in numerical methods for systems and control, but the field has strongly evolved since then. Examples of areas that were not in this volume but that are included in SLICOT are periodic systems, differential algebraic equations, and model reduction. Areas which still need new results and software are the control of large-scale systems, obtained either from discretizations of partial differential equations or from the interconnection of a large number of interacting systems. But it is unclear for the moment which will be the methods of choice for such problems. We still need to understand the numerical challenges in such areas, before we can propose numerically reliable software for these problems: the area is still quite open for new developments.

Cross-References

▶ Computer-Aided Control Systems Design: Introduction and Historical Overview

▶ Interactive Environments and Software Tools for CACSD

Bibliography

Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1995) LAPACK users' guide, 2nd edn. SIAM, Philadelphia. http://www.netlib.org/lapack/

Benner P, Laub AJ, Mehrmann V (1997) Benchmarks for the numerical solution of algebraic Riccati equations. Control Syst Mag 17:18–28

Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT-A subroutine library in systems and control theory. Appl Comput Control Signals Circuits 1:499–532

Demmel JW, Kågström B (1993) The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: robust software with error bounds and applications. Part I: theory and algorithms. ACM Trans Math Softw 19:160–174

Denham MJ, Benson CJ (1981) Implementation and documentation standards for the software library in control engineering (SLICE). Technical report 81/3, Kingston Polytechnic, Control Systems Research Group, Kingston

Dongarra JJ, Du Croz J, Duff IS, Hammarling S (1990) A set of level 3 basic linear algebra subprograms. ACM Trans Math Softw 16:1–17

Frederick DK (1988) Benchmark problems for computer aided control system design. In: Proceedings of the 4th IFAC symposium on computer-aided control systems design, Bejing, pp 1–6

Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, Baltimore

Gomez C, Bunks C, Chancelior J-P, Delebecque F (1997) Integrated scientific computing with scilab. Birkhäuser, Boston. https://www.scilab.org/

Grübel G (1983) Die regelungstechnische Programmmbibliothek RASP. Regelungstechnik 31: 75–81

Luenberger DG (1967) Canonical forms for linear multivariable systems. IEEE Trans Autom Control 12(3):290–293

Paige CC (1981) Properties of numerical algorithms related to computing controllability. IEEE Trans Autom Control AC-26:130–138

Patel R, Laub A, Van Dooren P (eds) (1994) Numerical linear algebra techniques for systems and control. IEEE, Piscataway

The Control and Systems Library SLICOT (2012) The NICONET society. NICONET e.V. http://www.niconet-ev.info/en/

The MathWorks, Inc. (2013) MATLAB version 8.1. The MathWorks, Inc., Natick

The Numerical Algorithms Group (1993) NAG SLICOT library manual, release 2. The Numerical Algorithms Group, Wilkinson House, Oxford. Updates Release 1 of May 1990

The Working Group on Software (1996) SLICOT implementation and documentation standards 2.1. WGS-report 96-1. http://www.icm.tu-bs.de/NICONET/reports.html

Van Dooren P (1981) The generalized eigenstructure problem in linear system theory. IEEE Trans Autom Control AC-26:111–129

Varga A (ed) (2004) Special issue on numerical awareness in control. Control Syst Mag 24-1: 14–17

Wieslander J (1977) Scandinavian control library. A subroutine library in the field of automatic control. Technical report, Department of Automatic Control, Lund Institute of Technology, Lund

Bilinear Control of Schrödinger PDEs

Karine Beauchard[1] and Pierre Rouchon[2]
[1]CNRS, CMLS, Ecole Polytechnique, Palaiseau, France
[2]Centre Automatique et Systèmes, Mines ParisTech, Paris Cedex 06, France

Abstract

This entry is an introduction to modern issues about controllability of Schrödinger PDEs with bilinear controls. This model is pertinent for a quantum particle, controlled by an electric field. We review recent developments in the field, with discrimination between exact and approximate controllabilities, in finite or infinite time. We also underline the variety of mathematical tools used by various teams in the last decade. The results are illustrated on several classical examples.

Keywords

Introduction

A quantum particle, in a space with dimension N ($N = 1, 2, 3$), in a potential $V = V(x)$, and in an electric field $u = u(t)$, is represented by a wave function $\psi : (t, x) \in \mathbb{R} \times \Omega \to \mathbb{C}$ on the $L^2(\Omega, \mathbb{C})$ sphere $\mathcal{S}$

$$\int_\Omega |\psi(t, x)|^2 \, dx = 1, \quad \forall t \in \mathbb{R},$$

where $\Omega \subset \mathbb{R}^N$ is a possibly unbounded open domain. In first approximation, the time evolution of the wave function is given by the Schrödinger equation,

$$\begin{cases} i \, \partial_t \psi(t, x) = (-\Delta + V) \, \psi(t, x) \\ -u(t)\mu(x) \, \psi(t, x), \ t \in (0, +\infty), x \in \Omega, \\ \psi(t, x) = 0, \ x \in \partial\Omega \end{cases} \tag{1}$$

where μ is the dipolar moment of the particle and $\hbar = 1$ here. Sometimes, this equation is considered in the more abstract framework

$$i \frac{d}{dt} \psi = (H_0 + u(t) H_1) \, \psi \tag{2}$$

where ψ lives on the unit sphere of a separable Hilbert space $\mathcal{H}$ and the Hamiltonians H_0, H_1 are Hermitian operators on $\mathcal{H}$. A natural question, with many practical applications, is the existence of a control u that steers the wave function ψ from a given initial state ψ_0, to a prescribed target ψ_f.

The goal of this survey is to present well-established results concerning exact and approximate controllabilities for the bilinear control system (1), with applications to relevant examples. The main difficulties are the infinite

dimension of $\mathcal{H}$ and the nonlinearity of the control system.

Preliminary Results

When the Hilbert space $\mathcal{H}$ has finite dimension n, then controllability of Eq. (2) is well understood (D'Alessandro 2008). If, for example, the Lie algebra spanned by H_0 and H_1 coincides with $u(n)$, the set of skew-Hermitian matrices, then system (2) is globally controllable: for any initial and final states $\psi_0, \psi_f \in \mathcal{H}$ of length one, there exist $T > 0$ and a bounded open-loop control $[0, T] \ni t \mapsto u(t)$ steering ψ from $\psi(0) = \psi_0$ to $\psi(T) = \psi_f$.

In infinite dimension, this idea served to intuit a negative controllability result in Mirrahimi and Rouchon (2004), but the above characterization cannot be generalized because iterated Lie brackets of unbounded operators are not necessarily well defined. For example, the quantum harmonic oscillator

$$i \, \partial_t \psi \, (t, x) = - \, \partial_x^2 \psi \, (t, x) + x^2 \psi \, (t, x)$$
$$- u \, (t) \, x \psi \, (t, x) \, , \quad x \in \mathbb{R}, \quad (3)$$

is not controllable (in any reasonable sense) (Mirrahimi and Rouchon 2004) even if all its Galerkin approximations are controllable (Fu et al. 2001). Thus, much care is required in the use of Galerkin approximations to prove controllability in infinite dimension. This motivates the search of different methods to study exact controllability of bilinear PDEs of form (1).

In infinite dimension, the norms need to be specified. In this article, we use Sobolev norms. For $s \in \mathbb{N}$, the Sobolev space $H^s(\Omega)$ is the space of functions $\psi : \Omega \to \mathbb{C}$ with square integrable derivatives $d^k \psi$ for $k = 0, \ldots, s$ (derivatives are well defined in the distribution sense). $H^s(\Omega)$ is endowed with the norm $\| \psi \|_{H^s} := \left(\sum_{k=0}^s \| d^k \psi \|_{L^2(\Omega)}^2 \right)^{1/2}$. We also use the space $H_0^1(\Omega)$ which contains functions $\psi \in H^1(\Omega)$ that vanish on the boundary $\partial \Omega$ (in the trace sense) (Brézis 1999).

The first control result of the literature states the noncontrollability of system (1) in $(H^2 \cap H_0^1)$ $(\Omega) \cap \mathcal{S}$ with controls $u \in L^2((0, T), \mathbb{R})$ (Ball et al. 1982; Turinici 2000). More precisely, by applying $L^2(0, T)$ controls u, the reachable wave functions $\psi(T)$ form a subset of $(H^2 \cap H_0^1)$ $(\Omega) \cap \mathcal{S}$ with empty interior. This statement does not give obstructions for system (1) to be controllable in different functional spaces as we will see below, but it indicates that controllability issues are much more subtle in infinite dimension than in finite dimension.

Local Exact Controllability

In 1D and with Discrete Spectrum
This section is devoted to the 1D PDE:

$$\begin{cases} i \, \partial_t \psi \, (t, x) = - \partial_x^2 \psi \, (t, x) \\ - u(t) \mu \, (x) \, \psi \, (t, x) \, , \quad x \in (0, 1) \, , t \in (0, T) \, , \\ \psi \, (t, 0) = \psi \, (t, 1) = 0. \end{cases}$$
$$(4)$$

We call "ground state" the solution of the free system ($u = 0$) built with the first eigenvalue and eigenvector of $-\partial_x^2$: $\psi_1(t, x) = \sqrt{2} \, \sin(\pi x) \, e^{-i \pi^2 t}$. Under appropriate assumptions on the dipolar moment μ, then system (4) is controllable around the ground state, locally in $H_{(0)}^3 \, (0, 1) \cap \mathcal{S}$, with controls in $L^2((0, T), \mathbb{R})$, as stated below.

Theorem 1 *Assume $\mu \in H^3((0, 1), \mathbb{R})$ and*

$$\left| \int_0^1 \mu \, (x) \sin \, (\pi x) \sin \, (k \pi x) \, dx \right| \geq \frac{c}{k^3}, \, \forall k \in \mathbb{N}^*$$
$$(5)$$

for some constant $c > 0$. Then, for every $T > 0$, there exists $\delta > 0$ such that for every $\psi_0, \psi_f \in \mathcal{S} \cap H_{(0)}^3 \, ((0, 1), \mathbb{C})$ with $\| \psi_0 - \psi_1(0) \|_{H^3} + \| \psi_f - \psi_1(T) \|_{H^3} < \delta$, there exists $u \in L^2((0, T), \mathbb{R})$ such that the solution of (4) with initial condition $\psi(0, x) = \psi_0(x)$ satisfies $\psi(T) = \psi_f$.

Here, $H_{(0)}^3 \, (0, 1) := \{ \psi \in H^3((0, 1), \mathbb{C}); \psi = \psi'' = 0 \text{ at } x = 0, 1 \}$. We refer to Beauchard

and Laurent (2010) and Beauchard et al. (2013) for proof and generalizations to nonlinear PDEs. The proof relies on the linearization principle, by applying the classical inverse mapping theorem to the endpoint map. Controllability of the linearized system around the ground state is a consequence of assumption (5) and classical results about trigonometric moment problems. A subtle smoothing effect allows to prove C^1 regularity of the endpoint map.

The assumption (5) holds for generic $\mu \in H^3((0,1), \mathbb{R})$ and plays a key role for local exact controllability to hold in *small* time T. In Beauchard and Morancey (2014), local exact controllability is proved under the weaker assumption, namely, $\mu'(0) \pm \mu'(1) \neq 0$, but only in *large* time T.

Moreover, under appropriate assumptions on μ, references Coron (2006) and Beauchard and Morancey (2014) propose explicit motions that are impossible in small time T, with small controls in L^2. Thus, a positive minimal time is required for local exact controllability, even if information propagates at infinite speed. This minimal time is due to nonlinearities; its characterization is an open problem.

Actually, assumption $\mu'(0) \pm \mu'(1) \neq 0$ is not necessary for local exact controllability in large time. For instance, the quantum box, i.e.,

$$\begin{cases} i\,\partial_t \psi\,(t,x) = -\partial_x^2 \psi\,(t,x) \\ -u(t)x\psi\,(t,x)\,, \quad x \in (0,1)\,, \\ \psi\,(t,0) = \psi\,(t,1) = 0, \end{cases} \quad (6)$$

is treated in Beauchard (2005). Of course, these results are proved with additional techniques: power series expansions and Coron's return method (Coron 2007).

There is no contradiction between the negative result of section "Preliminary Results" and the positive result of Theorem 1. Indeed, the wave function cannot be steered between any two points ψ_0, ψ_f of $H^2 \cap H_0^1$, but it can be steered between any two points ψ_0, ψ_f of $H_{(0)}^3$, which is smaller than $H^2 \cap H_0^1$. In particular, $H_{(0)}^3\,((0,1), \mathbb{C})$ has an empty interior in $H_{(0)}^2\,((0,1), \mathbb{C})$. Thus, there is no incompatibility between the reachable set to have empty interior

in $H^2 \cap H_0^1$ and the reachable set to coincide with $H_{(0)}^3$.

Open Problems in Multi-D or with Continuous Spectrum

The linearization principle used to prove Theorem 1 does not work in multi-D: the trigonometric moment problem, associated to the controllability of the linearized system, cannot be solved. Indeed, its frequencies, which are the eigenvalues of the Dirichlet Laplacian operator, do not satisfy a required gap condition (Loreti and Komornik 2005).

The study of a toy model (Beauchard 2011) suggests that if local controllability holds in 2D (with a priori bounded L^2-controls) then a positive minimal time is required, whatever μ is. The appropriate functional frame for such a result is an open problem.

In 3D or in the presence of continuous spectrum, we conjecture that local exact controllability does not hold (with a priori bounded L^2-controls) because the gap condition in the spectrum of the Dirichlet Laplacian operator is violated (see Beauchard et al. (2010) for a toy model from nuclear magnetic resonance and ensemble controllability as originally stated in Li and Khaneja (2009)). Thus, exact controllability should be investigated with controls that are not a priori bounded in L^2; this requires new techniques. We refer to Nersesyan and Nersisyan (2012a) for precise negative results.

Finally, we emphasize that exact controllability in multi-D but in *infinite* time has been proved in Nersesyan and Nersisyan (2012a,b), with techniques similar to one used in the proof of Theorem 1.

Approximate Controllability

Different approaches have been developed to prove approximate controllability.

Lyapunov Techniques
Due to measurement effect and back action, closed-loop controls in the Schrödinger frame are

not appropriate. However, closed-loop controls may be computed via numerical simulations and then applied to real quantum systems in open loop, without measurement. Then, the strategy consists in designing damping feedback laws, thanks to a controlled Lyapunov function, which encodes the distance to the target. In finite dimension, the convergence proof relies on LaSalle invariance principle. In infinite dimension, this principle works when the trajectories of the closed-loop system are compact (in the appropriate space), which is often difficult to prove. Thus, two adaptations have been proposed: approximate convergence (Beauchard and Mirrahimi 2009; Mirrahimi 2009) and weak convergence (Beauchard and Nersesyan 2010) to the target.

Variational Methods and Global Exact Controllability

The global approximate controllability of (1), in any Sobolev space, is proved in Nersesvan (2010), under generic assumptions on (V, μ), with Lyapunov techniques and variational arguments.

Theorem 2 *Let $V, \mu \in C^\infty(\overline{\Omega}, \mathbb{R})$ and $(\lambda_j)_{j \in N*}, (\phi_j)_{j \in N*}$ be the eigenvalues and normalized eigenvectors of $(-\Delta + V)$. Assume $\langle \mu \phi_j, \phi_1 \rangle \neq 0$, for all $j \geq 2$ and $\lambda_1 - \lambda_j \neq \lambda_p - \lambda_q$ for all $j, p, q \in \mathbb{N}*$ such that $\{1, j\} \neq \{p, q\}; j \neq 1$. Then, for every $s > 0$, the system (1) is globally approximately controllable in $H^s_{(V)}: D\left[(-\Delta + V)^{s/2}\right]$, the domain of $(-\Delta + V)^{s/2}$: for every $\epsilon, \delta > 0$ and $\psi_0 \in \mathcal{S} \cap H^s_{(V)}$, there exist a time $T > 0$ and a control $u \in C_0^\infty((0, T), \mathbb{R})$ such that the solution of (1) with initial condition $\psi(0) = \psi_0$ satisfies $\|\psi(T) - \phi_1\|_{H^{s-\delta}_{(V)}} < \epsilon$.*

This theorem is of particular importance. Indeed, in 1D and for appropriate choices of (V, μ), *global* exact controllability of (1) in H^{3+} can be proved by combining the following:

- Global approximate controllability in H^3 given by Theorem 2,
- Local exact controllability in H^3 given by Theorem 1,

- Time reversibility of the Schrödinger equation (i.e., if $(\psi(t, x), u(t))$ is a trajectory, then so is $(\psi^*(T - t, x), u(T - t))$ where ψ^* is the complex conjugate of ψ).

Let us expose this strategy on the quantum box (6). First, one can check the assumptions of Theorem 2 with $V(x) = \gamma x$ and $\mu(x) = (1-\gamma)x$ when $\gamma > 0$ is small enough. This means that, in (6), we consider controls $u(t)$ of the form $\gamma + u(t)$. Thus, an initial condition $\psi_0 \in H^{3+}_{(0)}$ can be steered arbitrarily close to the first eigenvector $\varphi_{1,\gamma}$ of $(-\partial_x^2 + \gamma x)$, in H^3 norm. Moreover, by a variant of Theorem 1, the local exact controllability of (6) holds in $H^3_{(0)}$ around $\phi_{1,\gamma}$. Therefore, the initial condition $\psi_0 \in H^{3+}_{(0)}$ can be steered exactly to $\phi_{1,\gamma}$ in finite time. By the time reversibility of the Schrödinger equation, we can also steer exactly the solution from $\phi_{1,\gamma}$ to any target $\psi_f \in H^{3+}$. Therefore, the solution can be steered exactly from any initial condition $\psi_0 \in H^{3+}_{(0)}$ to any target $\psi_f \in H^{3+}_{(0)}$ in finite time.

Geometric Techniques Applied to Galerkin Approximations

In Boscain et al. (2012, 2013) and Chambrion et al. (2009) the authors study the control of Schrödinger PDEs, in the abstract form (2) and under technical assumptions on the (unbounded) operators H_0 and H_1 that ensure the existence of solutions with piecewise constant controls u:

1. H_0 is skew-adjoint on its domain $D(H_0)$.
2. There exists a Hilbert basis $(\varphi_k)_{k \in N}$ of $\mathcal{H}$ made of eigenvectors of $H_0: H_0 \phi_k = i\lambda_k \phi_k$ and $\phi_k \in D(H_1), \forall k \in \mathbb{N}$.
3. $H_0 + uH_1$ is essentially skew-adjoint (not necessarily with domain $D(H_0)$) for every $u \in [0, \delta]$ for some $\delta > 0$.
4. $\langle H_1 \varphi_j, \varphi_k \rangle = 0$ for every $j, k \in \mathbb{N}$ such that $\lambda_j = \lambda_k$ and $j \neq k$.

Theorem 3 *Assume that, for every $j, k \in \mathbb{N}$, there exists a finite number of integers $p_1, \ldots, p_r \in \mathbb{N}$ such that*

$$p_1 = j, \ p_r = k, \ \langle H_1 \varphi_{p_l}, \varphi_{p_{l+1}} \rangle$$
$$\neq 0, \forall l = 1, \ldots, r - 1$$

$|\lambda_L - \lambda_M| \neq |\lambda_{p_l} - \lambda_{p_{l+1}}|, \forall 1 \leq l \leq r - 1$, $LM \in \mathbb{N}$ with $\{L, M\} \neq \{p_l, p_{l+1}\}$.

Then for every $\epsilon > 0$ and ψ_0, ψ_f in the unit sphere of $\mathcal{H}$, there exists a piecewise constant function $u : [0, T_\epsilon] \to [0, \delta]$ such that the solution of (2) with initial condition $\psi(0) = \psi_0$ satisfies $\|\psi(T_\epsilon) - \psi_f\|_{\mathcal{H}} < \epsilon$.

We refer to Boscain et al. (2012, 2013) and Chambrion et al. (2009) for proof and additional results such as estimates on the L^1 norm of the control. Note that H_0 is not necessarily of the form $(-\Delta + V)$, H_1 can be unbounded, δ may be arbitrary small, and the two assumptions are generic with respect to (H_0, H_1). The connectivity and transition frequency conditions in Theorem 3 mean physically that each pair of H_0 eigenstates is connected via a finite number of first-order (one-photon) transitions and that the transition frequencies between pairs of eigenstates are all different.

Note that, contrary to Theorems 2, Theorem 3 cannot be combined with Theorem 1 to prove global exact controllability. Indeed, functional spaces are different: $\mathcal{H} = L^2(\Omega)$ in Theorem 3, whereas H^3-regularity is required for Theorem 1.

This kind of results applies to several relevant examples such as the control of a particle in a quantum box by an electric field (6) and the control of the planar rotation of a linear molecule by means of two electric fields:

$$i \, \partial_t \psi(t, \theta) = \left(-\partial_0^2 + u_1(t) \cos(\theta) + u_2(t) \sin(\theta)\right) \psi(t, \theta), \quad \theta \in \mathbb{T}$$

where $\mathbb{T}$ is the lD-torus. However, several other systems of physical interest are not covered by these results such as trapped ions modeled by two coupled quantum harmonic oscillators. In Ervedoza and Puel (2009), specific methods have been used to prove their approximate controllability.

Concluding Remarks

The variety of methods developed by different authors to characterize controllability of Schrödinger PDEs with bilinear control is the sign of a rich structure and subtle nature of control issues. New methods will probably be necessary to answer the remaining open problems in the field.

This survey is far from being complete. In particular, we do not consider numerical methods to derive the steering control such as those used in NMR (Nielsen et al. 2010) to achieve robustness versus parameter uncertainties or such as monotone algorithms (Baudouin and Salomon 2008; Liao et al. 2011) for optimal control (Cancès et al. 2000). We do not consider also open quantum systems where the state is then the density operator ρ, a nonnegative Hermitian operator with unit trace on H. The Schrödinger equation is then replaced by the Lindblad equation:

$$\frac{d}{dt}\rho = -\iota \left[H_0 + uH_1, \rho\right] + \sum_\nu L_\nu \rho L_\nu^\dagger$$
$$-\frac{1}{2}\left(L_\nu^\dagger L_\nu \rho + \rho L_\nu^\dagger L_\nu\right)$$

with operator L_ν related to the decoherence channel ν. Even in the case of finite dimensional Hilbert space H, controllability of such system is not yet well understood and characterized (see Altafini (2003) and Kurniawan et al. (2012)).

Cross-References

▶ Control of Quantum Systems
▶ Robustness Issues in Quantum Control

Bibliography

Altafini C (2003) Controllability properties for finite dimensional quantum Markovian master equations. J Math Phys 44(6):2357–2372

Ball JM, Marsden JE, Slemrod M (1982) Controllability for distributed bilinear systems. SIAM J Control Optim 20:575–597

Baudouin L, Salomon J (2008) Constructive solutions of a bilinear control problem for a Schrödinger equation. Syst Control Lett 57(6):453–464

Beauchard K (2005) Local controllability of a 1-D Schrödinger equation. J Math Pures Appl 84:851–956

Beauchard K (2011) Local controllability and non controllability for a ID wave equation with bilinear control. J Diff Equ 250:2064–2098

Beauchard K, Laurent C (2010) Local controllability of 1D linear and nonlinear Schrödinger equations with bilinear control. J Math Pures Appl 94(5):520–554

Beauchard K, Mirrahimi M (2009) Practical stabilization of a quantum particle in a one-dimensional infinite square potential well. SIAM J Control Optim 48(2):1179–1205

Beauchard K, Morancey M (2014) Local controllability of 1D Schrödinger equations with bilinear control and minimal time, vol 4. Mathematical Control and Related Fields

Beauchard K, Nersesyan V (2010) Semi-global weak stabilization of bilinear Schrödinger equations. CRAS 348(19–20):1073–1078

Beauchard K, Coron J-M, Rouchon P (2010) Controllability issues for continuous spectrum systems and ensemble controllability of Bloch equations. Commun Math Phys 290(2):525–557

Beauchard K, Lange H, Teismann H (2013, preprint) Local exact controllability of a Bose-Einstein condensate in a 1D time-varying box. arXiv:1303.2713

Boscain U, Caponigro M, Chambrion T, Sigalotti M (2012) A weak spectral condition for the controllability of the bilinear Schrödinger equation with application to the control of a rotating planar molecule. Commun Math Phys 311(2):423–455

Boscain U, Chambrion T, Sigalotti M (2013) On some open questions in bilinear quantum control. arXiv:1304.7181

Brézis H (1999) Analyse fonctionnelles: théorie et applications. Dunod, Paris

Cancès E, Le Bris C, Pilot M (2000) Contrôle optimal bilinéaire d'une équation de Schrödinger. CRAS Paris 330:567–571

Chambrion T, Mason P, Sigalotti M, Boscain M (2009) Controllability of the discrete-spectrum Schrödinger equation driven by an external field. Ann Inst Henri Poincaré Anal Nonlinéaire 26(l):329–349

Coron J-M (2006) On the small-time local controllability of a quantum particle in a moving one-dimensional infinite square potential well. C R Acad Sci Paris I 342:103–108

Coron J-M (2007) Control and nonlinearity. Mathematical surveys and monographs, vol 136. American Mathematical Society, Providence

D'Alessandro D (2008) Introduction to quantum control and dynamics. Applied mathematics and nonlinear science. Chapman & Hall/CRC, Boca Raton

Ervedoza S, Puel J-P (2009) Approximate controllability for a system of Schrödinger equations modeling a single trapped ion. Ann Inst Henri Poincaré Anal Nonlinéaire 26(6):2111–2136

Fu H, Schirmer SG, Solomon AI (2001) Complete controllability of finite level quantum systems. J Phys A 34(8):1678–1690

Kurniawan I, Dirr G, Helmke U (2012) Controllability aspects of quantum dynamics: unified approach for closed and open systems. IEEE Trans Autom Control 57(8):1984–1996

Li JS, Khaneja N (2009) Ensemble control of Bloch equations. IEEE Trans Autom Control 54(3):528–536

Liao S-K, Ho T-S, Chu S-I, Rabitz HH (2011) Fast-kick-off monotonically convergent algorithm for searching optimal control fields. Phys Rev A 84(3):031401

Loreti P, Komornik V (2005) Fourier series in control theory. Springer, New York

Mirrahimi M (2009) Lyapunov control of a quantum particle in a decaying potential. Ann Inst Henri Poincaré (c) Nonlinear Anal 26:1743–1765

Mirrahimi M, Rouchon P (2004) Controllability of quantum harmonic oscillators. IEEE Trans Autom Control 49(5):745–747

Nersesvan V (2010) Global approximate controllability for Schrödinger equation in higher Sobolev norms and applications. Ann IHP Nonlinear Anal 27(3):901–915

Nersesyan V, Nersisyan H (2012a) Global exact controllability in infinite time of Schrödinger equation. J Math Pures Appl 97(4):295–317

Nersesyan V, Nersisyan H (2012b) Global exact controllability in infinite time of Schrödinger equation: multidimensional case. Preprint: arXiv:1201.3445

Nielsen NC, Kehlet C, Glaser SJ and Khaneja N (2010) Optimal Control Methods in NMR Spectroscopy. eMagRes

Turinici G (2000) On the controllability of bilinear quantum systems. In: Le Bris C, Defranceschi M (eds) Mathematical models and methods for ab initio quantum chemistry. Lecture notes in chemistry, vol 74. Springer

Boundary Control of 1-D Hyperbolic Systems

Georges Bastin[1] and Jean-Michel Coron[2]
[1]Department of Mathematical Engineering, University Catholique de Louvain, Louvain-La-Neuve, Belgium
[2]Laboratoire Jacques-Louis Lions, University Pierre et Marie Curie, Paris, France

Abstract

One-dimensional hyperbolic systems are commonly used to describe the evolution of various physical systems. For many of these systems, controls are available on the boundary. There

are then two natural questions: controllability (steer the system from a given state to a desired target) and stabilization (construct feedback laws leading to a good behavior of the closed loop system around a given set point).

Keywords

Chromatography; Controllability; Electrical lines; Hyperbolic systems; Open channels; Road traffic; Stabilization

One-Dimensional Hyperbolic Systems

The operation of many physical systems may be represented by hyperbolic systems in one space dimension. These systems are described by the following partial differential equation:

$$Y_t + A(Y)Y_x = 0, \ \ t \in [0, T], \ \ x \in [0, L], \ (1)$$

where:
- t and x are two independent variables: a time variable $t \in [0, T]$ and a space variable $x \in [0, L]$ over a finite interval.
- $Y : [0, T] \times [0, L] \rightarrow \mathbb{R}^n$ is the vector of state variables.
- $A : \mathbb{R}^n \rightarrow \mathcal{M}_{n,n}(\mathbb{R})$ with $\mathcal{M}_{n,n}(\mathbb{R})$ is the set of $n \times n$ real matrices.
- Y_t and Y_x denote the partial derivatives of Y with respect to t and x, respectively.

The system (1) is **hyperbolic** which means that $A(Y)$ has n distinct real eigenvalues (called characteristic velocities) for all Y in a domain of $\mathbb{R}^n$. Here are some typical examples of physical models having the form of a hyperbolic system.

Electrical Lines

First proposed by Heaviside in (1885, 1886 and 1887), the equations of (lossless) electrical lines (also called telegrapher equations) describe the propagation of current and voltage along electrical transmission lines (see Fig. 1). It is a hyperbolic system of the following form:

$$\begin{pmatrix} I_t \\ V_t \end{pmatrix} + \begin{pmatrix} 0 & L_s^{-1} \\ C_s^{-1} & 0 \end{pmatrix} \begin{pmatrix} I_x \\ V_x \end{pmatrix} = 0, \quad (2)$$

where $I(t, x)$ is the current intensity, $V(t, x)$ is the voltage, L_s is the self-inductance per unit length, and C_s is the self-capacitance per unit length. The system has two characteristic velocities (which are the eigenvalues of the matrix A):

$$\lambda_1 = \frac{1}{\sqrt{L_s C_s}} > 0 > \lambda_2 = -\frac{1}{\sqrt{L_s C_s}}. \quad (3)$$

Saint-Venant Equation for Open Channels

First proposed by Barré de Saint-Venant in (1871), the Saint-Venant equations (also called *shallow water equations*) describe the propagation of water in open channels (see Fig. 2). In the case of a horizontal channel with rectangular cross section, unit width, and negligible friction, the Saint-Venant model is a hyperbolic system of the form

$$\begin{pmatrix} H_t \\ V_t \end{pmatrix} + \begin{pmatrix} V & H \\ g & V \end{pmatrix} \begin{pmatrix} H_x \\ V_x \end{pmatrix} = 0, \quad (4)$$

where $H(t, x)$ is the water depth, $V(t, x)$ is the water horizontal velocity, and g is the gravity acceleration. Under subcritical flow conditions, the system is hyperbolic with characteristic velocities

$$\lambda_1 = V + \sqrt{gH} > 0 > \lambda_2 = V - \sqrt{gH}. \quad (5)$$

### Aw-Rascle Equations for Fluid Models of Road Traffic

In the fluid paradigm for road traffic modeling, the traffic is described in terms of two basic macroscopic state variables: the density $\varrho(t, x)$ and the speed $V(t, x)$ of the vehicles at position x along the road at time t. The following dynamical model for road traffic was proposed by Aw and Rascle in (2000):

$$\begin{pmatrix} \varrho_t \\ V_t \end{pmatrix} + F(Y) \begin{pmatrix} V & \varrho \\ 0 & V - Q(\varrho) \end{pmatrix} \begin{pmatrix} \varrho_t \\ V_t \end{pmatrix} = 0. \quad (6)$$
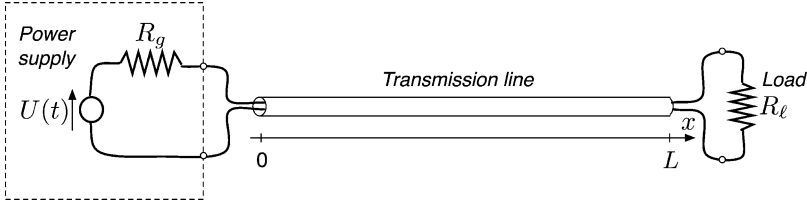
Boundary Control of 1-D Hyperbolic Systems, Fig. 1 Transmission line connecting a power supply to a resistive load R_ℓ; the power supply is represented by a Thevenin equivalent with *efm* $U(t)$ and internal resistance R_g

Boundary Control of 1-D Hyperbolic Systems, Fig. 2 Lateral view of a pool of a horizontal open channel
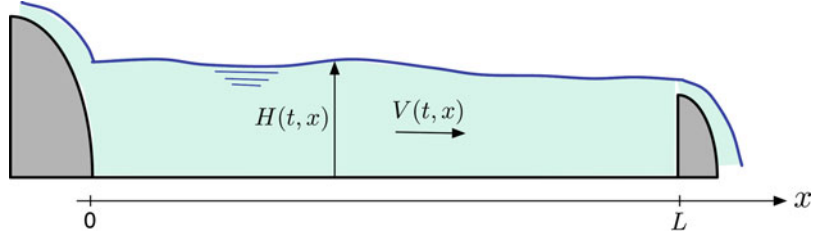
The system is hyperbolic with characteristic velocities

$$\lambda_1 = V \; > \; \lambda_2 = V - Q(\varrho). \qquad (7)$$

In this model the first equation of (6) is a continuity equation that represents the conservation of the number of vehicles on the road. The second equation of (6) is a phenomenological model describing the speed variations induced by the driver's behavior.

Chromatography

In chromatography, a mixture of species with different affinities is injected in the carrying fluid at the entrance of the process as illustrated in Fig. 3. The various substances travel at different propagation speeds and are ultimately separated in different bands. The dynamics of the mixture are described by a system of partial differential equations:

$$(P_i + L_i(P))_t + V(P_i)_x = 0 \; \; i = 1, \ldots, n,$$

$$L_i(P) = \frac{k_i P_i}{1 + \sum_j k_j P_j / P_{\max}}, \qquad (8)$$
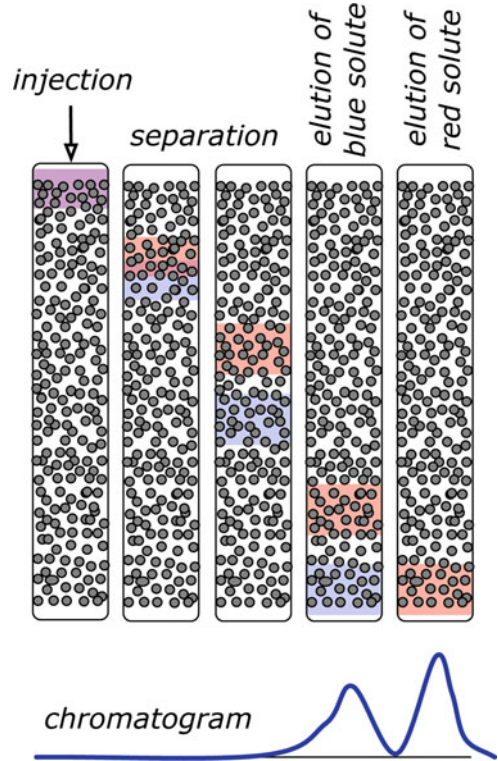
Boundary Control of 1-D Hyperbolic Systems, Fig. 3 Principle of chromatography

where P_i $(i = 1, \ldots, n)$ denote the densities of the n carried species. The function $L_i(P)$

(called the "Langmuir isotherm") was proposed by Langmuir in (1916).

Boundary Control

Boundary control of 1-D hyperbolic systems refers to situations where manipulated control inputs are physically located at the boundaries. Formally, this means that the system (1) is considered under n boundary conditions having the general form

$$B\big(Y(t,0), Y(t,L), U(t)\big) = 0, \qquad (9)$$

with $B : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}^n$. The dependence of the map B on $(Y(t,0), Y(t,L))$ refers to natural physical constraints on the system. The function $U(t) \in \mathbb{R}^q$ represents a set of q exogenous control inputs. The following examples illustrate how the control boundary conditions (9) may be defined for some commonly used control devices:

1. **Electrical lines.** For the circuit represented in Fig. 1, the line model (2) is to be considered under the following boundary conditions:

$$V(t,0) + R_g I(t,0) = U(t),$$
$$V(t,L) - R_\ell I(t,L) = 0.$$

   The telegrapher equations (2) coupled with these boundary conditions constitute therefore a boundary control system with the voltage $U(t)$ as control input.

2. **Open channels.** A standard situation is when the boundary conditions are assigned by tunable hydraulic gates as in irrigation canals and navigable rivers; see Fig. 4.

   The hydraulic model of mobile spillways gives the boundary conditions

$$H(t,0)V(t,0) = k_G \sqrt{\big[Z_0(t) - U_0(t)\big]^3},$$
$$H(t,L)V(t,L) = k_G \sqrt{\big[H(t,L) - U_L(t)\big]^3},$$

   where $H(t,0)$ and $H(t,L)$ denote the water depth at the boundaries inside the pool, $Z_0(t)$ and $Z_L(t)$ are the water levels on the other side of the gate, k_G is a constant gate shape parameter, and U_0 and U_L represent the weir elevations. The Saint-Venant equations coupled to these boundary conditions constitute a boundary control system with $U_0(t)$ and $U_L(t)$ as command signals.

3. **Ramp metering.** Ramp metering is a strategy that uses traffic lights to regulate the flow of traffic entering freeways according to measured traffic conditions as illustrated in Fig. 5. For the stretch of motorway represented in this figure, the boundary conditions are

$$\varrho(t,0)V(t,0) = Q_{\text{in}}(t) + U(t),$$
$$\varrho(t,L)V(t,L) = Q_{\text{out}}(t),$$

   where $U(t)$ is the inflow rate controlled by the traffic lights. The Aw-Rascle equations (6) coupled to these boundary conditions constitute a boundary control system with $U(t)$ as
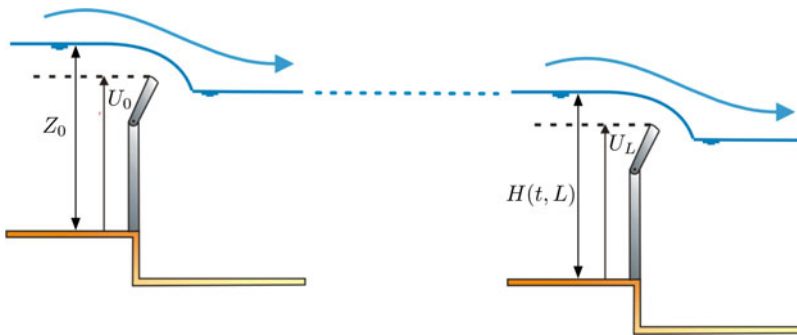
Boundary Control of 1-D Hyperbolic Systems, Fig. 4 Hydraulic gates at the input and the output of a pool
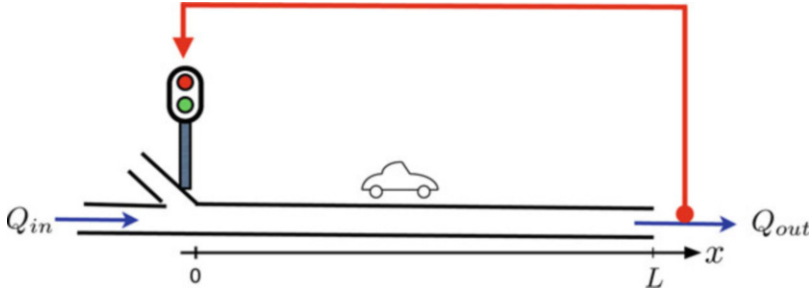
Boundary Control of 1-D Hyperbolic Systems, Fig. 5 Ramp metering on a stretch of a motorway

the command signal. In a feedback implementation of the ramp metering strategy, $U(t)$ may be a function of the measured disturbances $Q_{\text{int}}(t)$ or $Q_{\text{out}}(t)$ that are imposed by the traffic conditions.

4. **Simulated moving bed chromatography** is a technology where several interconnected chromatographic columns are switched periodically against the fluid flow. This allows for a continuous separation with a better performance than the discontinuous single-column chromatography. An efficient operation of SMB chromatography requires a tight control of the process by manipulating the inflow rates in the columns. This process is therefore a typical example of a periodic boundary control hyperbolic system.

Controllability

In this section and in the following one, $Y^* \in \mathbb{R}^n$ is such that none of the eigenvalues of $A(Y^*)$ are 0. After an appropriate linear state transformation, the matrix $A(Y^*)$ can be assumed to be diagonal, with distinct and nonzero entries:

$$A(Y^*) = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad (10)$$

$$\lambda_1 > \lambda_2 > \cdots > \lambda_m > 0 > \lambda_{m+1} > \cdots > \lambda_n.$$

Let $Y^+ \in \mathbb{R}^m$ and $Y^- \in \mathbb{R}^{n-m}$ be such that $Y^{\mathrm{T}} = (Y^{+\mathrm{T}} \ Y^{-\mathrm{T}})^{\mathrm{T}}$.

For the boundary control system (1), (9), the local controllability issue is to investigate if, starting from a given initial state $Y_0 : x \in$ $[0, L] \mapsto Y_0(x) \in \mathbb{R}^n$, it is possible to reach in time T a desired target state $Y_1 : x \in [0, L] \mapsto Y_1(x) \in \mathbb{R}^n$, with $Y_0(x)$ and $Y_1(x)$ close to Y^*.

Theorem 4 (See Li and Rao 2003) *If there exist control inputs $U^+(t)$ and $U^-(t)$ such that the boundary conditions* (9) *are equivalent to*

$$Y^+(t, 0) = U^+(t), \quad Y^-(t, L) = U^-(t), \tag{11}$$

then the boundary control system (1)*,* (11) *is locally controllable for the C^1-norm if and only if $T > T_c$ with*

$$T_c = \max \left\{ \frac{L}{|\lambda_1|}, \ldots, \frac{L}{|\lambda_n|} \right\}.$$

Feedback Stabilization

For the boundary control system (1), (9), the problem of local boundary feedback stabilization is the problem of finding boundary feedback control actions

$$U(t) = F(Y(t, 0), Y(t, L), Y^*),$$

$$F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^p, \tag{12}$$

such that the system trajectory exponentially converges to a desired steady-state Y^* (called *set point*) from any initial condition $Y_0(x)$ close to Y^*. In such case, the set point is said to be exponentially stable.

Theorem 5 (See Coron et al. 2008) *If there exists a boundary feedback $U(t) =$*

$F(Y(t, 0), Y(t, L), Y^*)$ *such that the boundary conditions* (9) *are written in the form*

$$\begin{pmatrix} Y^+(t, 0) \\ Y^-(t, L) \end{pmatrix} = G \begin{pmatrix} Y^+(t, L) \\ Y^-(t, 0) \end{pmatrix}, \quad G(Y^*) = Y^*, \tag{13}$$

then, for the boundary control system (1)*,* (13)*, the set point Y^* is locally exponentially stable for the H^2-norm if*

$$Inf \{\|\Delta G'(Y^*)\Delta^{-1}\|; \ \Delta \in \mathcal{D}\} < 1,$$

where $\| \ \|$ denotes the usual 2-norm of $n \times n$ real matrices, $G'(Y^*)$ denotes the Jacobian matrix of the map G at Y^*, and $\mathcal{D}$ denotes the set of $n \times n$ diagonal real matrices with strictly positive diagonal entries.

For the stabilization in the C^1-norm, another sufficient condition is given in Li (1994).

Summary and Future Directions

With suitable boundary controls, hyperbolic systems can be controlled and stabilized around a desired set point. However, in many situations the hyperbolic model is not sufficient: one needs to add a zero-order term and (1) has to be replaced by

$$Y_t + A(Y)Y_x + C(Y) = 0, \ t \in [0, T], \ x \in [0, L], \tag{14}$$

where $C : \mathbb{R}^n \to \mathbb{R}^n$. This is, for example, the case for the open channels when slope and friction cannot be neglected. Note that the set point Y^* may now depend on x. For the controllability issue, the new term $C(Y)$ turns out to be not essential; see in particular Li (2010). The situation is not the same for the stabilization and only partial results are known. In particular, Coron et al. (2013) uses Krstic's backstepping approach (Krstic and Smyshlyaev 2008) to treat the case $n = 2$ and $m = 1$.

Another important issue for the system (1) is the observability problem: assume that the state is measured on the boundary during the interval of time $[0, T]$, can one recover the initial data?

As shown in Li (2010), this problem has strong connections with the controllability problem and the system (1) is observable if the time T is large enough.

The above results are on smooth solutions of (1). However, the system (1) is known to be well posed in class of BV-solutions (Bounded Variations), with extra conditions (e.g., entropy type); see in particular Bressan (2000). There are partial results on the controllability in this class. See, in particular, Ancona and Marson (1998) and Horsin (1998) for $n = 1$. For $n = 2$, it is shown in Bressan and Coclite (2002) that Theorem 4 no longer holds in general in the BV class. However, there are positive results for important physical systems; see, for example, Glass (2007) for the 1-D isentropic Euler equation.

Cross-References

▶ Controllability and Observability
▶ Control of Fluids and Fluid-Structure Interactions
▶ Control of Linear Systems with Delays
▶ Feedback Stabilization of Nonlinear Systems
▶ Lyapunov's Stability Theory

Bibliography

Ancona F, Marson A (1998) On the attainable set for scalar nonlinear conservation laws with boundary control. SIAM J Control Optim 36(1):290–312 (electronic)

Aw A, Rascle M (2000) Resurrection of "second order" models of traffic flow. SIAM J Appl Math 60(3):916–938 (electronic)

Barré de Saint-Venant A-C (1871) Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit. Comptes rendus de l'Académie des Sciences de Paris, Série 1, Mathématiques, 53:147–154

Bressan A (2000) Hyperbolic systems of conservation laws. Volume 20 of Oxford lecture series in mathematics and its applications. Oxford University Press, Oxford. The one-dimensional Cauchy problem

Bressan A, Coclite GM (2002) On the boundary control of systems of conservation laws. SIAM J Control Optim 41(2):607–622 (electronic)

Coron J-M, Bastin G, d'Andréa Novel B (2008) Dissipative boundary conditions for one-dimensional

nonlinear hyperbolic systems. SIAM J Control Optim 47(3):1460–1498

Coron J-M, Vazquez R, Krstic M, Bastin G (2013) Local exponential H^2 stabilization of a 2×2 quasilinear hyperbolic system using backstepping. SIAM J Control Optim 51(3):2005–2035

Glass O (2007) On the controllability of the 1-D isentropic Euler equation. J Eur Math Soc (JEMS) 9(3):427–486

Heaviside O (1885, 1886 and 1887) Electromagnetic induction and its propagation. The Electrician, reprinted in Electrical Papers, 2 vols, London. Macmillan and co. 1892

Horsin T (1998) On the controllability of the Burgers equation. ESAIM Control Optim Calc Var 3:83–95 (electronic)

Krstic M, Smyshlyaev A (2008) Boundary control of PDEs. Volume 16 of advances in design and control. Society for Industrial and Applied Mathematics (SIAM), Philadelphia. A course on backstepping designs

Langmuir I (1916) The constitution and fundamental properties of solids and liquids. Part I. Solids. J Am Chem Soc 38:2221–2295

Li T (1994) Global classical solutions for quasilinear hyperbolic systems. Volume 32 of RAM: research in applied mathematics. Masson, Paris

Li T (2010) Controllability and observability for quasilinear hyperbolic systems. Volume 3 of AIMS series on applied mathematics. American Institute of Mathematical Sciences (AIMS), Springfield

Li T, Rao B-P (2003) Exact boundary controllability for quasi-linear hyperbolic systems. SIAM J Control Optim 41(6):1748–1755 (electronic)

Boundary Control of Korteweg-de Vries and Kuramoto–Sivashinsky PDEs

Eduardo Cerpa
Departamento de Matemática, Universidad Técnica Federico Santa María, Valparaiso, Chile

Abstract

The Korteweg-de Vries (KdV) and the Kuramoto-Sivashinsky (KS) partial differential equations are used to model nonlinear propagation of one-dimensional phenomena. The KdV equation is used in fluid mechanics to describe waves propagation in shallow water surfaces, while the KS equation models front propagation in reaction-diffusion systems. In this article, the boundary control of these equations is considered

when they are posed on a bounded interval. Different choices of controls are studied for each equation.

Keywords

Controllability; Dispersive equations; Higher-order partial differential equations; Parabolic equations; Stabilizability

Introduction

The Korteweg-de Vries (KdV) and the Kuramoto-Sivashinsky (KS) equations have very different properties because they do not belong to the same class of partial differential equations (PDEs). The first one is a third-order nonlinear dispersive equation

$$y_t + y_x + y_{xxx} + y y_x = 0, \qquad (1)$$

and the second one is a fourth-order nonlinear parabolic equation

$$u_t + u_{xxxx} + \lambda u_{xx} + u u_x = 0, \qquad (2)$$

where $\lambda > 0$ is called the anti-diffusion parameter. However, they have one important characteristic in common. They are both used to model nonlinear propagation phenomena in the space x-direction when the variable t stands for time. The KdV equation serves as a model for waves propagation in shallow water surfaces (Korteweg and de Vries 1895), and the KS equation models front propagation in reaction-diffusion phenomena including some instability effects (Kuramoto and Tsuzuki 1975; Sivashinsky 1977).

From a control point of view, a new common characteristic arises. Because of the order of the spatial derivatives involved, when studying these equations on a bounded interval $[0, L]$, two boundary conditions have to be imposed at the same point, for instance, at $x = L$. Thus, we can consider control systems where we control one boundary condition but not all the boundary data at one endpoint of the interval. This

configuration is not possible for the classical wave and heat equations where at each extreme, only one boundary condition exists and therefore controlling one or all the boundary data at one point is the same.

The KdV equation being of third order in space, three boundary conditions have to be imposed: one at the left endpoint $x = 0$ and two at the right endpoint $x = L$. For the KS equation, four boundary conditions are needed to get a well-posed system, two at each extreme. We will focus on the cases where Dirichlet and Neumann boundary conditions are considered because lack of controllability phenomena appears. This holds for some special values of the length of the interval for the KdV equation and depends on the anti-diffusion coefficient λ for the KS equation.

The particular cases where the lack of controllability occurs can be seen as isolated anomalies. However, those phenomena give us important information on the systems. In particular, any method independent of the value of those constants cannot control or stabilize the system when acting from the corresponding control input where trouble appears. In all of these cases, for both the KdV and the KS equations, the space of uncontrollable states is finite dimensional, and therefore, some methods coming from the control of ordinary differential equations can be applied.

General Definitions

Infinite-dimensional control systems described by PDEs have attracted a lot of attention since the 1970s. In this framework, the state of the control system is given by the solution of an evolution PDE. This solution can be seen as a trajectory in an infinite-dimensional Hilbert space H, for instance, the space of square integrable functions or some Sobolev space. Thus, for any time t, the state belongs to H. Concerning the control input, this is either an internal force distributed in the domain, or a punctual force localized within the domain, or some boundary data as considered in this article. For any time t, the control belongs to a control space U, which can be, for instance,

the space of bounded functions. The main control properties to be mentioned in this article are controllability, stability, and stabilization. A control system is said to be exactly controllable if the system can be driven from any initial state to another one in finite time. This kind of properties holds, for instance, for hyperbolic system as the wave equation. The notion of null-controllability means that the system can be driven to the origin from any initial state. The main example for this property is the heat equation, which presents regularizing effects. Even if the initial data is discontinuous, right after $t = 0$, the solution of the heat equation becomes very smooth, and therefore, it is not possible to impose a discontinuous final state. A system is said to be asymptotically stable if the solutions of the system without any control converge as the time goes to infinity to a stationary solution of the PDE. When this convergence holds with a control depending at each time on the state of the system (feedback control), the system is said to be stabilizable by means of a feedback control law.

All these properties have local versions when a smallness condition for the initial and/or the final state is added. This local character is normally due to the nonlinearity of the system.

The KdV Equation

The classical approach to deal with nonlinearities is first to linearize the system around a given state or trajectory, then to study the linear system and finally to go back to the nonlinear one by means of an inversion argument or a fixed-point theorem. Linearizing (1) around the origin, we get the equation

$$y_t + y_x + y_{xxx} = 0, \tag{3}$$

which can be studied on a finite interval $[0, L]$ under the following three boundary conditions:

$$y(t, 0) = h_1(t), \quad y(t, L) = h_2(t), \quad \text{and}$$
$$y_x(t, L) = h_3(t). \tag{4}$$

Thus, viewing $h_1(t), h_2(t), h_3(t) \in \mathbb{R}$ as controls and the solution $y(t, \cdot) : [0, L] \to \mathbb{R}$ as the state, we can consider the linear control system (3)–(4) and the nonlinear one (1)–(4).

We will report on the role of each input control when the other two are off. The tools used are mainly the duality controllability-observability, Carleman estimates, the multiplier method, the compactness-uniqueness argument, the backstepping method, and fixed-point theorems. Surprisingly, the control properties of the system depend strongly on the location of the controls.

Theorem 1 *The linear KdV system* (3)–(4) *is:*

1. *Null-controllable when controlled from h_1 (i.e., $h_2 = h_3 = 0$) (Glass and Guerrero 2008).*
2. *Exactly controllable when controlled from h_2 (i.e., $h_1 = h_3 = 0$) if and only if L does not belong to a set O of critical lengths defined in Glass and Guerrero (2010).*
3. *Exactly controllable when controlled from h_3 (i.e., $h_1 = h_2 = 0$) if and only if L does not belong to a set of critical lengths N defined in Rosier (1997).*
4. *Asymptotically stable to the origin if $L \notin N$ and no control is applied (Perla Menzala et al. 2002).*
5. *Stabilizable by means of a feedback law using h_1 only (i.e., $h_2 = h_3 = 0$) Cerpa and Coron (2013).*

If $L \in N$ or $L \in O$, one says that L is a critical length since the linear control system (3)–(4) loses controllability properties when only one control input is applied. In those cases, there exists a finite-dimensional subspace of $L^2(0, L)$ which is unreachable from 0 for the linear system. The sets N and O contain infinitely many critical lengths, but they are countable sets.

When one is allowed to use more than one boundary control input, there is no critical spatial domain, and the exact controllability holds for any $L > 0$. This is proved in Zhang (1999) when three boundary controls are used. The case of two control inputs is solved in Rosier (1997), Glass and Guerrero (2010), and Cerpa et al. (2013).

Previous results concern the linearized control system. Considering the nonlinearity yy_x, we obtain the original KdV control system and the following results.

Theorem 2 *The nonlinear KdV system* (1)–(4) *is:*

1. *Locally null-controllable when controlled from h_1 (i.e., $h_2 = h_3 = 0$) (Glass and Guerrero 2008).*
2. *Locally exactly controllable when controlled from h_2 (i.e., $h_1 = h_3 = 0$) if L does not belong to the set O of critical lengths (Glass and Guerrero 2010).*
3. *Locally exactly controllable when controlled from h_3 (i.e., $h_1 = h_2 = 0$). If L belongs to the set of critical lengths N, then a minimal time of control may be required (see Cerpa 2014).*
4. *Asymptotically stable to the origin if $L \notin N$ and no control is applied (Perla Menzala et al. 2002).*
5. *Locally stabilizable by means of a feedback law using h_1 only (i.e., $h_2 = h_3 = 0$) (Cerpa and Coron 2013).*

Item 3 in Theorem 2 is a truly nonlinear result obtained by applying a power series method introduced in Coron and Crépeau (2004). All other items are implied by perturbation arguments based on the linear control system. The related control system formed by (1) with boundary controls

$$y(t, 0) = h_1(t), \quad y_x(t, L) = h_2(t), \quad \text{and}$$
$$y_{xx}(t, L) = h_3(t), \tag{5}$$

is studied in Cerpa et al. (2013), and the same phenomenon of critical lengths appears.

The KS Equation

Applying the same strategy than for KdV, we linearize (2) around the origin to get the equation

$$u_t + u_{xxxx} + \lambda u_{xx} = 0, \tag{6}$$

which can be studied on the finite interval $[0, 1]$ under the following four boundary conditions:

$$u(t, 0) = v_1(t), \quad u_x(t, 0) = v_2(t),$$
$$u(t, 1) = v_3(t), \quad \text{and} \quad u_x(t, 1) = v_4(t). \tag{7}$$

Thus, viewing $v_1(t), v_2(t), v_3(t), v_4(t) \in \mathbb{R}$ as controls and the solution $u(t, \cdot) : [0, 1] \to \mathbb{R}$ as the state, we can consider the linear control system (6)–(7) and the nonlinear one (2)–(7). The role of the parameter λ is crucial. The KS equation is parabolic and the eigenvalues of system (6)–(7) with no control ($v_1 = v_2 = v_3 = v_4 = 0$) go to $-\infty$. If λ increases, then the eigenvalues move to the right. When $\lambda > 4\pi^2$, the system becomes unstable because there are a finite number of positive eigenvalues. In this unstable regime, the system loses control properties for some values of λ.

Theorem 3 *The linear KS control system* (6)–(7) *is:*
1. *Null-controllable when controlled from v_1 and v_2 (i.e., $v_3 = v_4 = 0$). The same is true when controlling v_3 and v_4 (i.e., $v_1 = v_2 = 0$) (Cerpa and Mercado 2011; Lin Guo 2002).*
2. *Null-controllable when controlled from v_2 (i.e., $v_1 = v_2 = v_3 = 0$) if and only if λ does not belong to a countable set M defined in Cerpa (2010).*
3. *Asymptotically stable to the origin if $\lambda < 4\pi^2$ and no control is applied (Liu and Krstic 2001).*
4. *Stabilizable by means of a feedback law using v_2 only (i.e., $v_2 = v_3 = v_4 = 0$) if and only if $\lambda \notin M$ (Cerpa 2010).*

In the critical case $\lambda \in M$, the linear system is not null-controllable anymore if we control v_2 only (item 2 in Theorem 3). The space of noncontrollable states is finite dimensional. To obtain the null-controllability of the linear system in these cases, we have to add another control. Controlling with v_2 and v_4 does not improve the situation in the critical cases. Unlike that, the system becomes null-controllable if we can act on v_1 and v_2. This result with two input controls has been proved in Lin Guo (2002) for the case $\lambda = 0$ and in Cerpa and Mercado (2011) in the general case (item 1 in Theorem 3).

It is known from Liu and Krstic (2001) that if $\lambda < 4\pi^2$, then the system is exponentially stable in $L^2(0, 1)$. On the other hand, if $\lambda = 4\pi^2$, then zero becomes an eigenvalue of the system, and therefore the asymptotic stability fails. When $\lambda > 4\pi^2$, the system has positive eigenvalues and becomes unstable. In order to stabilize this system, a finite-dimensional-based feedback law can be designed by using the pole placement method (item 4 in Theorem 3).

Previous results concern the linearized control system. If we add the nonlinearity uu_x, we obtain the original KS control system and the following results.

Theorem 4 *The KS control system* (2)–(7) *is:*
1. *Locally null-controllable when controlled from v_1 and v_2 (i.e., $v_3 = v_4 = 0$). The same is true when controlling v_3 and v_4 (i.e., $v_1 = v_2 = 0$) (Cerpa and Mercado 2011).*
2. *Asymptotically stable to the origin if $\lambda < 4\pi^2$ and no control is applied (Liu and Krstic 2001).*

There are less results for the nonlinear systems than for the linear one. This is due to the fact that the spectral techniques used to study the linear system with only one control input are not robust enough to deal with perturbations in order to address the nonlinear control system.

Summary and Future Directions

The KdV and the KS equations possess both noncontrol results when one boundary control input is applied. This is due to the fact that both are higher-order equations, and therefore, when posed on a bounded interval, more than one boundary condition should be imposed at the same point. The KdV equation is exactly controllable when acting from the right and null-controllable when acting from the left. On the other hand, the KS equation, being parabolic as the heat equation, is not exactly controllable but null-controllable. Most of the results are implied by the behaviors of the corresponding linear system, which are very well understood.

For the KdV equation, the main directions to investigate at this moment are the controllability and the stability for the nonlinear equation in critical domains. Among others, some questions concerning controllability, minimal time of control, and decay rates for the stability are open. Regarding the KS equation, there are few results for the nonlinear system with one control input even if we are not in a critical value of the anti-diffusion parameter. In the critical cases, the controllability and stability issues are wide open.

In general, for PDEs, there are few results about delay phenomena, output feedback laws, adaptive control, and other classical questions in control theory. The existing results on these topics mainly concern the more popular heat and wave equations. As KdV and KS equations are one dimensional in space, many mathematical tools are available to tackle those problems. For all that, to our opinion, the KdV and KS equations are excellent candidates to continue investigating these control properties in a PDE framework.

Cross-References

- ▶ Boundary Control of 1-D Hyperbolic Systems
- ▶ Controllability and Observability
- ▶ Control of Fluids and Fluid-Structure Interactions
- ▶ Feedback Stabilization of Nonlinear Systems
- ▶ Stability: Lyapunov, Linear Systems

Recommended Reading

The book Coron (2007) is a very good reference to study the control of PDEs. In Cerpa (2014), a tutorial presentation of the KdV control system is given. Control system for PDEs with boundary conditions and internal controls is considered in Rosier and Zhang (2009) and the references therein for the KdV equation and in Armaou and Christofides (2000) and Christofides and Armaou (2000) for the KS equation. Control topics as delay and adaptive control are studied in the framework of PDEs in Krstic (2009) and Smyshlyaev and Krstic (2010), respectively.

Bibliography

Armaou A, Christofides PD (2000) Feedback control of the Kuramoto-Sivashinsky equation. Physica D 137:49–61

Cerpa E (2010) Null controllability and stabilization of a linear Kuramoto-Sivashinsky equation. Commun Pure Appl Anal 9:91–102

Cerpa E (2014) Control of a Korteweg-de Vries equation: a tutorial. Math Control Rel Fields 4:45–99

Cerpa E, Coron J-M (2013) Rapid stabilization for a Korteweg-de Vries equation from the left dirichlet boundary condition. IEEE Trans Autom Control 58:1688–1695

Cerpa E, Mercado A (2011) Local exact controllability to the trajectories of the 1-D Kuramoto-Sivashinsky equation. J Differ Equ 250:2024–2044

Cerpa E, Rivas I, Zhang B-Y (2013) Boundary controllability of the Korteweg-de Vries equation on a bounded domain. SIAM J Control Optim 51:2976–3010

Christofides PD, Armaou A (2000) Global stabilization of the Kuramoto-Sivashinsky equation via distributed output feedback control. Syst Control Lett 39:283–294

Coron JM (2007) Control and nonlinearity. American Mathematical Society, Providence

Coron J-M, Crépeau E (2004) Exact boundary controllability of a nonlinear KdV equation with critical lengths. J Eur Math Soc 6:367–398

Glass O, Guerrero S (2008) Some exact controllability results for the linear KdV equation and uniform controllability in the zero-dispersion limit. Asymptot Anal 60:61–100

Glass O, Guerrero S (2010) Controllability of the KdV equation from the right Dirichlet boundary condition. Syst Control Lett 59:390–395

Korteweg DJ, de Vries G (1895) On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. Philos Mag 39:422–443

Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhauser, Boston

Kuramoto Y, Tsuzuki T (1975) On the formation of dissipative structures in reaction-diffusion systems. Theor Phys 54:687–699

Lin Guo Y-J (2002) Null boundary controllability for a fourth order parabolic equation. Taiwan J Math 6: 421–431

Liu W-J, Krstic M (2001) Stability enhancement by boundary control in the Kuramoto-Sivashinsky equation. Nonlinear Anal Ser A Theory Methods 43: 485–507

Perla Menzala G, Vasconcellos CF, Zuazua E (2002) Stabilization of the Korteweg-de Vries equation with localized damping. Q Appl Math LX:111–129

Rosier L (1997) Exact boundary controllability for the Korteweg-de Vries equation on a bounded domain. ESAIM Control Optim Calc Var 2:33–55

Rosier L, Zhang B-Y (2009) Control and stabilization of the Korteweg-de Vries equation: recent progresses. J Syst Sci Complex 22:647–682

Sivashinsky GI (1977) Nonlinear analysis of hydrody-
    namic instability in laminar flames – I derivation of
    basic equations. Acta Astronaut 4:1177–1206
Smyshlyaev A, Krstic M (2010) Adaptive control of
    parabolic PDEs. Princeton University Press, Princeton
Zhang BY (1999) Exact boundary controllability of the
    Korteweg-de Vries equation. SIAM J Control Optim
    37:543–565

Bounds on Estimation

Arye Nehorai[1] and Gongguo Tang[2]
[1]Preston M. Green Department of Electrical and
Systems Engineering, Washington University in
St. Louis, St. Louis, MO, USA
[2]Department of Electrical Engineering &
Computer Science, Colorado School of Mines,
Golden, CO, USA

Abstract

We review several universal lower bounds on
statistical estimation, including deterministic
bounds on unbiased estimators such as Cramér-
Rao bound and Barankin-type bound, as well as
Bayesian bounds such as Ziv-Zakai bound. We
present explicit forms of these bounds, illustrate
their usage for parameter estimation in Gaussian
additive noise, and compare their tightness.

Keywords

Barankin-type bound; Cramér-Rao bound;
Mean-squared error; Statistical estimation;
Ziv-Zakai bound

Introduction

Statistical estimation involves inferring the values
of parameters specifying a statistical model from
data. The performance of a particular statistical
algorithm is measured by the error between the
true parameter values and those estimated by the
algorithm. However, explicit forms of estimation
error are usually difficult to obtain except for
the simplest statistical models. Therefore, per-
formance bounds are derived as a way of quan-
tifying estimation accuracy while maintaining
tractability.

In many cases, it is beneficial to quantify
performance using universal bounds that are in-
dependent of the estimation algorithms and rely
only upon the model. In this regard, universal
lower bounds are particularly useful as it provides
means to assess the difficulty of performing es-
timation for a particular model and can act as
benchmarks to evaluate the quality of any algo-
rithm: the closer the estimation error of the algo-
rithm to the lower bound, the better the algorithm.
In the following, we review three widely used
universal lower bounds on estimation: Cramér-
Rao bound (CRB), Barankin-type bound (BTB),
and Ziv-Zakai bound (ZZB). These bounds find
numerous applications in determining the per-
formance of sensor arrays, radar, and nonlinear
filtering; in benchmarking various algorithms;
and in optimal design of systems.

Statistical Model and Related Concepts

To formalize matters, we define a statistical
model for estimation as a family of parameterized
probability density functions in $\mathbb{R}^N$: $\{p(x; \theta) :
\theta \in \Theta \subset \mathbb{R}^d\}$. We observe a realization of
$x \in \mathbb{R}^N$ generated from a distribution $p(x; \theta)$,
where $\theta \in \Theta$ is the true parameter to be
estimated from data x. Though we assume a
single observation x, the model is general enough
to encompass multiple independent, identically
distributed samples (i.i.d.) by considering the
joint probability distribution.

An estimator of θ is a measurable function of
the observation $\hat{\theta}(x) : \mathbb{R}^N \rightarrow \Theta$. An unbiased
estimator is one such that

$$\mathbb{E}_\theta \left\{ \hat{\theta}(x) \right\} = \theta, \forall \theta \in \Theta. \quad (1)$$

Here we used the subscript θ to emphasize that the expectation is taken with respect to $p(x; \theta)$. We focus on the performance of unbiased estimators in this entry. There are various ways to measure the error of the estimator $\hat{\theta}(x)$. Two typical ones are the error covariance matrix:

$$\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} = \text{Cov}(\hat{\theta}), \quad (2)$$

where the equation holds only for unbiased estimators, and the mean-squared error (MSE):

$$\mathbb{E}_\theta \left\{ \|\hat{\theta}(x) - \theta\|_2^2 \right\} = \text{trace} \left(\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta) \right. \right.$$
$$\left. \left. (\hat{\theta} - \theta)^T \right\} \right). \quad (3)$$

Example 1 (Signal in additive Gaussian noise (SAGN)) To illustrate the usage of different estimation bounds, we use the following statistical model as a running example:

$$x_n = s_n(\theta) + w_n, n = 0, \dots, N - 1. \quad (4)$$

Here $\theta \in \Theta \subset \mathbb{R}$ is a scalar parameter to be estimated and the noise w_n follows i.i.d. Gaussian distribution with mean 0 and *known* variance σ^2. Therefore, the density function for x is

$$p(x; \theta)$$
$$= \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(x_n - s_n(\theta))^2}{2\sigma^2} \right\}$$
$$= \frac{1}{(\sqrt{2\pi}\sigma)^N} \exp \left\{ -\sum_{n=0}^{N-1} \frac{(x_n - s_n(\theta))^2}{2\sigma^2} \right\}.$$

In particular, we consider the frequency estimation problem where $s_n(\theta) = \cos(2\pi n\theta)$ with $\Theta = [0, \frac{1}{4})$.

Cramér-Rao Bound

The Cramér-Rao bound (CRB) (Kay 2001a; Stoica and Nehorai 1989; Van Trees 2001) is arguably the most well-known lower bounds on

estimation. Define the Fisher information matrix $I(\theta)$ via

$$I_{i,j}(\theta) = \mathbb{E}_\theta \left\{ \frac{\partial}{\partial \theta_i} \log p(x; \theta) \frac{\partial}{\partial \theta_j} \log p(x; \theta) \right\}$$
$$= -\mathbb{E}_\theta \left\{ \frac{\partial^2}{\partial \theta_i \partial \theta_j} \log p(x; \theta) \right\}.$$

Then for *any* unbiased estimator $\hat{\theta}$, the error covariance matrix is bounded by

$$\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} \succeq [I(\theta)]^{-1}, \quad (5)$$

where $A \succeq B$ for two symmetric matrices means $A - B$ is positive semidefinite. The inverse of the Fisher information matrix $\text{CRB}(\theta) = [I(\theta)]^{-1}$ is called the Cramér-Rao bound.

When θ is scalar, $I(\theta)$ measures the expected sensitivity of the density function with respect to changes in the parameter. A density family that is more sensitive to parameter changes (larger $I(\theta)$) will generate observations that look more different when the true parameter varies, making it easier to estimate (smaller error).

Example 2 For the SAGN model (4), the CRB is

$$\text{CRB}(\theta) = I(\theta)^{-1} = \frac{\sigma^2}{\sum_{n=0}^{N-1} \left[\frac{\partial s_n(\theta)}{\partial \theta} \right]^2}. \quad (6)$$

The inverse dependence on the ℓ_2 norm of signal derivative suggests that signals more sensitive to parameter change are easier to estimate.

For the frequency estimation problem with $s_n(\theta) = \cos(2\pi n\theta)$, the CRB as a function of θ is plotted in Fig. 1.

There are many modifications of the basic CRB such as the posterior CRB (Tichavsky et al. 1998; Van Trees 2001), the hybrid CRB (Rockah and Schultheiss 1987), the modified CRB (D'Andrea et al. 1994), the concentrated CRB (Hochwald and Nehorai 1994), and constrained CRB (Gorman and Hero 1990; Marzetta 1993; Stoica and Ng 1998). The posterior CRB takes into account the prior information of the parameters when they are modeled as random
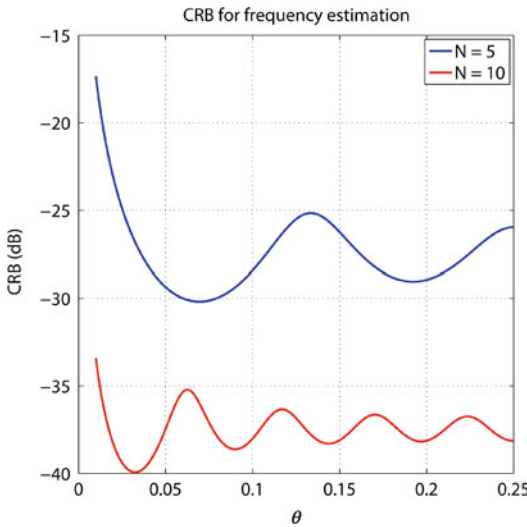
Bounds on Estimation, Fig. 1 Cramér-Rao bound on frequency estimation: $N = 5$ vs. $N = 10$

variables, while the hybrid CRB considers the case that the parameters contain both random and deterministic parts. The modified CRB and the concentrated CRB focus on handling nuisance parameters in a tractable manner. The application of the these CRBs requires a regular parameter space (e.g., an open set in $\mathbb{R}^d$). However, in many case, the parameter space Θ is a low-dimensional manifolds in $\mathbb{R}^d$ specified by equalities and inequalities. In this case, the constrained CRB provides tighter lower bounds by incorporating knowledge of the constraints.

Barankin Bound

CRB is a local bound in the sense that it involves only local properties (the first or second order derivatives) of the log-likelihood function. So if two families of log-likelihood functions coincide at a region near θ^0, the CRB at θ^0 would be the same, even if they are drastically different in other regions of the parameter space.

However, the entire parametric space should play a role in determining the difficulty of parameter estimation. To see this, imagine that there are two statistical models. In the first model there is another point $\theta^1 \in \Theta$ such that the likelihood

family $p(x; \theta)$ behaves similarly around θ^0 and θ^1, but these two points are not in neighborhoods of each other. Then it would be difficult to distinguish these two points for any estimation algorithm, and the estimation performance for the first statistical model would be bad (an extreme case is $p(x; \theta^0) \equiv p(x; \theta^1)$ in which case the model is non-identifiable; more discussions on identifiability and Fisher information matrix can be found in Hochwald and Nehorai (1997)). In the second model, we remove the point θ^1 and its near neighborhood from Θ, then the performance should get better. However, CRB for both models would remain the same whether we exclude θ^1 from Θ or not. As a matter of fact, CRB(θ^0) uses only the fact that the estimator is unbiased in a neighborhood of the true parameter θ^0.

Barankin bound addresses CRB's shortcoming of not respecting the global structure of the statistical model by introducing finitely many test points $\{\theta^i, i = 1, \ldots, M\}$ and ensures that the estimator is unbiased at the neighborhood of θ^0 as well as these test points (Forster and Larzabal 2002). The original Barankin bound (Barankin 1949) is derived for scalar parameter $\theta \in \Theta \subset \mathbb{R}$ and any unbiased estimator $\widehat{g(\theta)}$ for a function $g(\theta)$:

$$\mathbb{E}_\theta(\widehat{g(\theta)} - g(\theta))^2 \geq \sup_{M, \theta^i, a^i}$$

$$\frac{\left[\sum_{m=1}^M a^i (g(\theta^i) - g(\theta)) \right]^2}{\mathbb{E}_\theta \left[\sum_{m=1}^M a^i \frac{p(x; \theta^i)}{p(x; \theta)} \right]^2} \quad (7)$$
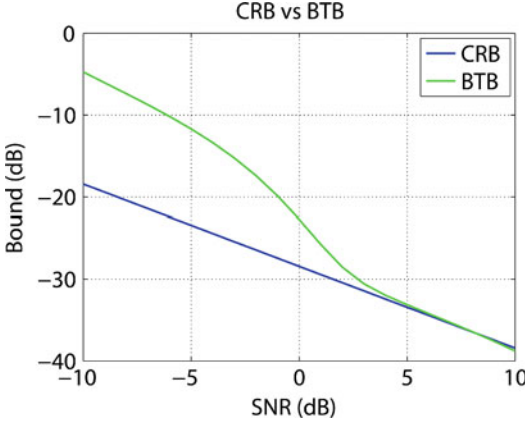
Using (7), we can derive a Barankin-type bound on the error covariance matrix of any unbiased estimator $\hat{\theta}(x)$ for a vector parameter $\theta \in \Theta \subset \mathbb{R}^d$ (Forster and Larzabal 2002):

$$\mathbb{E}_\theta \left\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \right\} \geq \Phi(B - 11^T)^{-1}\Phi^T, \quad (8)$$

where the matrices are defined via

$$B_{i,j} = \mathbb{E}_\theta \left\{ \frac{p(x; \theta^i)}{p(x; \theta)} \frac{p(x; \theta^j)}{p(x; \theta)} \right\}, 1 \leq i, j \leq M,$$

$$\Phi = \begin{bmatrix} \theta^1 - \theta \cdots \theta^M - \theta \end{bmatrix}$$

Bounds on Estimation, Fig. 2 Cramér-Rao bound vs. Barankin-type bound on frequency estimation when $\theta^0 = 0.1$. The BTB is obtained using $M = 10$ uniform random points

and 1 is the vector in $\mathbb{R}^M$ with all ones. Note that we have used θ^i with a superscript to denote different points in Θ, while θ_i with a subscript to denote the ith component of a point θ.

Since the bound (8) is valid for any M and any choice of test points $\{\theta^i\}$, we obtain the tightest bound by taking the supremum over all finite families of test points. Note that when we have d test points that approach θ in d linearly independent directions, the Barankin-type bound (8) converges to the CRB. If we have more than d test points, however, the Barankin-type bound is always not worse than the CRB. Particularly, the Barankin-type bound is much tighter in the regime of low signal-to-noise ratio (SNR) and small number of measurements, which allows one to investigate the "threshold" phenomena as shown in the next example.

Example 3 For the SAGN model, if we have M test points, the elements of matrix B are of the following form:

$$B_{i,j} = \exp\left\{\frac{1}{\sigma^2}\sum_{n=0}^{N-1}[s_n(\theta^i) - s_n(\theta)]\right.$$

$$\left.[s_n(\theta^j) - s_n(\theta)]\right\}$$

In most cases, it is extremely difficult to derive an analytical form of the Barankin bound by optimizing with respect to M and the test points $\{\theta^j\}$. In Fig. 2, we plot the Barankin-type bounds for $s_n(\theta) = \cos(2\pi n\theta)$ for $M = 10$ randomly selected test points. We observe that Barankin-type bound is tighter than the CRB when SNR is small. There is a SNR region around 0 dB that the Brankin-type bound drops drastically. This is usually called the "threshold" phenomenon. Practical systems operate much better in the region above the threshold.

The basic CRB and BTB belong to the family of deterministic "covariance inequality" bounds in the sense that the unknown parameter is assumed to be a *deterministic* quantity (as opposed to a *random* quantity). Additionally, both bounds work only for unbiased estimators, making them inappropriate performance indicators for biased estimators such as many regularization-based estimators.

Ziv-Zakai Bound

In this section, we introduce the Ziv-Zakai bound (ZZB) (Bell et al. 1997) that is applicable to any estimator (not necessarily unbiased). Unlike the CRB and BTB, the ZZB is a Bayesian bound and the errors are averaged by the prior distribution $p_\theta(\phi)$ of the parameter. For any $a \in \mathbb{R}^d$, the ZZB states that

$$a^T\mathbb{E}\left\{(\hat{\theta}(x) - \theta)(\hat{\theta}(x) - \theta)^T\right\}a \geq$$

$$\frac{1}{2}\int_0^\infty \mathcal{V}\left\{\max_{\delta:a^T\delta=h}\left[\int_{\mathbb{R}^d}(p_\theta(\phi) + p_\theta(\phi + \delta))\right.\right.$$

$$\left.\left. times P_{\min}(\phi, \phi + \delta)d\phi\right]\right\}hdh,$$

where the expectation is taken with respect to the joint disunity $p(x;\theta)p_\theta(\phi)$, $\mathcal{V}\{q(h)\} = \max_{r\geq 0}q(h + r)$ is the valley-filling function, and $P_{\min}(\phi, \phi + \delta)$ is the minimal probability of error for the following binary hypothesis testing problem:

$$H_0 : \theta = \phi; \qquad x \sim p(x;\phi)$$

$$H_1 : \theta = \phi + \delta; \; x \sim p(x; \phi + \delta)$$

with

$$\Pr(H_0) = \frac{p_\theta(\phi)}{p_\theta(\phi) + p_\theta(\phi + \delta)}$$

$$\Pr(H_1) = \frac{p_\theta(\phi + \delta)}{p_\theta(\phi) + p_\theta(\phi + \delta)}.$$

Example 4 For the ASGN model, we assume a uniform prior probability, i.e., $p_\theta(\phi) = 4, \phi \in [0, 1/4)$. The ZZB simplifies to

$$\mathbb{E}\left\{\|\hat{\theta}(x) - \theta\|_2^2\right\} \geq$$

$$\frac{1}{2} \int_0^{\frac{1}{4}} \mathcal{V}\left\{\left[\int_0^{\frac{1}{4}-h} 8 P_{\min}(\phi, \phi + h)d\phi\right]\right\} h\,dh.$$

The binary hypothesis testing problem is to decide which one of two signals is buried in additive Gaussian noise. The optimal detector with minimal probability of error is the minimum distance receiver (Kay 2001b), and the associated probability of error is

$$P_{\min}(\phi, \phi + h)$$

$$= Q\left(\frac{1}{2}\sqrt{\frac{\sum_{n=0}^{N-1}(s_n(\phi) - s_n(\phi + h))^2}{\sigma^2}}\right),$$

where $Q(h) = \int_h^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$. For the frequency estimation problem, we numerically estimate the integral and plot the resulting ZZB in Fig. 3 together with the mean-squared error for the maximum likelihood estimator (MLE).

Summary and Future Directions

We have reviewed several important performance bounds on statistical estimation problems, particularly, the Cramér-Rao bound, the Barankin-type bound, and the Ziv-Zakai bound. These bounds provide a universal way to quantify
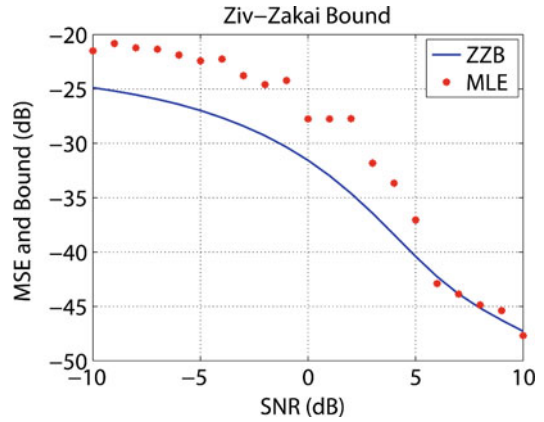
Bounds on Estimation, Fig. 3 Ziv-Zakai bound vs. maximum likelihood estimator for frequency estimation

the performance of statistically modeled physical systems that is independent of any specific algorithm.

Future directions of performance bounds on estimation include deriving tighter bounds, developing computational schemes to approximate existing bounds in a tractable way, and applying them to practical problems.

Cross-References

▶ Estimation, Survey on
▶ Particle Filters

Recommended Reading

Kay SM (2001a), Chapter 2 and 3; Stoica P, Nehorai A (1989); Van Trees HL (2001), Chapter 2.7; Forster and Larzabal (2002); Bell et al. (1997).

Bibliography

Barankin EW (1949) Locally best unbiased estimates. Ann Math Stat 20(4):477–501

Bell KL, Steinberg Y, Ephraim Y, Van Trees HL (1997) Extended Ziv-Zakai lower bound for vector parameter estimation. IEEE Trans Inf Theory 43(2):624–637

D'Andrea AN, Mengali U, Reggiannini R (1994) The modified Cramér-Rao bound and its application to synchronization problems. IEEE Trans Commun 42(234):1391–1399

Forster P, Larzabal P (2002) On lower bounds for deterministic parameter estimation. In: IEEE international conference on acoustics, speech, and signal processing (ICASSP), 2002, Orlando, vol 2. IEEE, pp II–1141

Gorman JD, Hero AO (1990) Lower bounds for parametric estimation with constraints. IEEE Trans Inf Theory 36(6):1285–1301

Hochwald B, Nehorai A (1994) Concentrated Cramér-Rao bound expressions. IEEE Trans Inf Theory 40(2):363–371

Hochwald B, Nehorai A (1997) On identifiability and information-regularity in parametrized normal distributions. Circuits Syst Signal Process 16(1):83–89

Kay SM (2001a) Fundamentals of statistical signal processing, volume 1: estimation theory. Prentice Hall, Upper Saddle River, NJ

Kay SM (2001b) Fundamentals of statistical signal processing, volume 2: detection theory. Prentice Hall, Upper Saddle River, NJ

Marzetta TL (1993) A simple derivation of the constrained multiple parameter Cramér-Rao bound. IEEE Trans Signal Process 41(6):2247–2249

Rockah Y, Schultheiss PM (1987) Array shape calibration using sources in unknown locations – part I: far-field sources. IEEE Trans Acoust Speech Signal Process 35(3):286–299

Stoica P, Nehorai A (1989) MUSIC, maximum likelihood, and Cramér-Rao bound. IEEE Trans Acoust Speech Signal Process 37(5):720–741

Stoica P, Ng BC (1998) On the Cramér-Rao bound under parametric constraints. IEEE Signal Process Lett 5(7):177–179

Tichavsky P, Muravchik CH, Nehorai A (1998) Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. IEEE Trans Signal Process 46(5):1386–1396

Van Trees HL (2001) Detection, estimation, and modulation theory: part 1, detection, estimation, and linear modulation theory. Jhon Wiley & Sons, Hoboken, NJ

BSDE

Building Control Systems

James E. Braun
Purdue University, West Lafayette, IN, USA

Abstract

This entry provides an overview of systems and issues related to providing optimized controls for commercial buildings. It includes a description of the evolution of the control systems over time, typical equipment and control variables, typical two-level hierarchal structure for feedback and supervisory control, definition of the optimal supervisory control problem, references to typical heuristic control approaches, and a description of current and future developments.

Keywords

Building automation systems (BAS); Cooling plant optimization; Energy management and controls systems (EMCS); Intelligent building controls

Introduction

Computerized control systems were developed in the 1980s for commercial buildings and are typically termed energy management and control systems (EMCS) or building automation systems (BAS). They have been most successfully applied to large commercial buildings that have hundreds of building zones and thousands of control points. Less than about 15 % of commercial buildings have EMCS, but they serve about 40 % of the floor area. Small commercial buildings tend not to have an EMCS, although there is a recent trend towards the use of wireless thermostats with cloud-based energy management solutions.

EMCS architectures for buildings have evolved from centralized to highly distributed systems as depicted in Fig. 1 in order to reduce
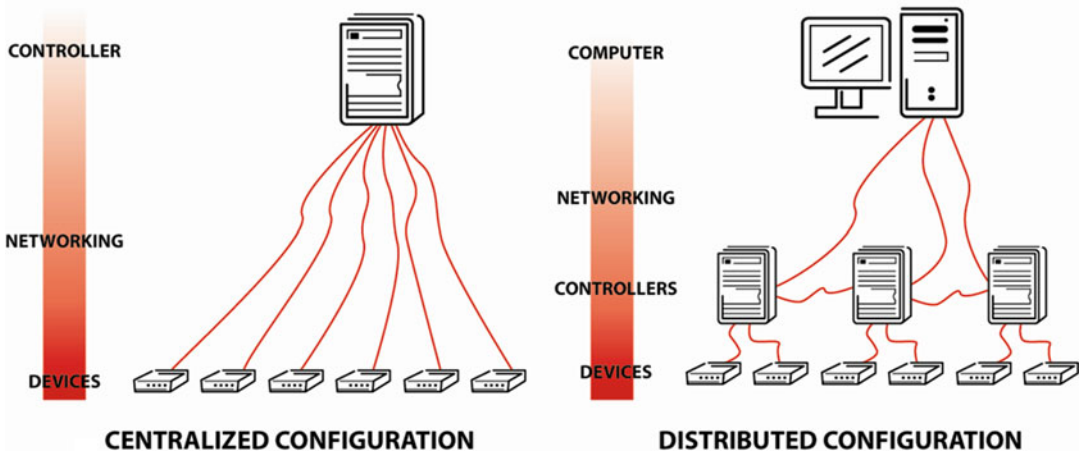
Building Control Systems, Fig. 1 Evolution from centralized to distributed network architectures

wiring costs and provide more modular solutions. The development of open communications protocols, such as BACNet, has enabled the use of distributed control devices from different vendors and improved the cost-effectiveness of ECMS. There has also been a recent trend towards the use of existing enterprise networks to reduce system installed costs and to more easily allow remote access and control from any Internet accessible device.

An EMCS for a large commercial building can automate the control of many of the building and system functions, including scheduling of lights and zone thermostat settings according to occupancy patterns. Security and fire safety systems tend to be managed using separate systems. In addition to scheduling, an EMCS manages the control of individual equipment and subsystems that provide heating, ventilation, and air conditioning of the building (HVAC). This control is achieved using a two-level hierarchical structure of local-loop and supervisory control. Local-loop control of individual set points is typically implemented using individual proportional-integral (PI) feedback algorithms that manipulate individual actuators in response to deviations from the set points. For example, supply air temperature from a cooling coil is controlled by adjusting a valve opening that provides chilled water to the coil. The second level of supervisory control specifies the set points and other modes of operation that depend on time and external conditions.

Each local-loop feedback controller acts independently, but their performance can be coupled to other local-loop controllers if not tuned appropriately. Adaptive tuning algorithms have been developed in recent years to enable controllers to automatically adjust to changing weather and load conditions. There are typically a number of degrees of freedom in adjusting supervisory control set points over a wide range while still achieving adequate comfort conditions. Optimal control of supervisory set points involves minimizing a cost function with respect to the free variables and subject to constraints. Although model-based, control optimization approaches are not typically employed in buildings, they have been used to inform the development and assessment of some heuristic control strategies. Most commonly, strategies for adjusting supervisory control variables are established at the control design phase based on some limited analysis of the HVAC system and specified as a sequence of operations that is programmed into the EMCS.

Systems, Equipment, and Controls

The greatest challenges and opportunities for optimizing supervisory control variables exist for
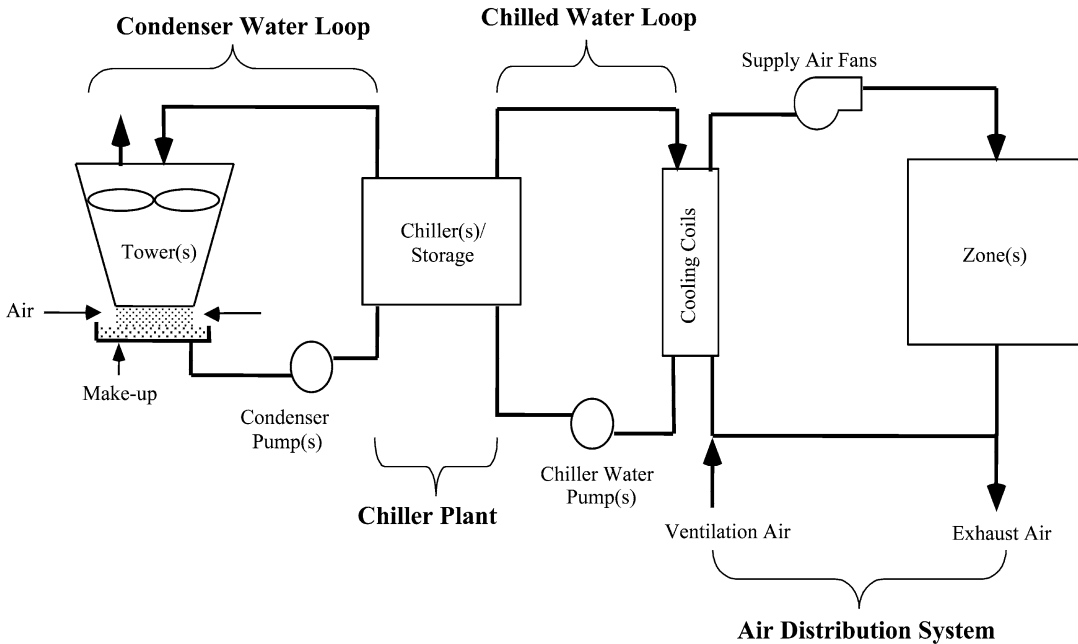
Building Control Systems, Fig. 2 Schematic of a chilled water cooling system

centralized cooling systems that are employed in large commercial buildings because of the large number of control variables and degrees of freedom along with utility rate incentives. A simplified schematic of a typical centralized cooling plant is shown in Fig. 2 with components grouped under air distribution, chilled water loop, chiller plant, and condenser water loop.

Typical **air distribution systems** include VAV (variable-air volume) boxes within the zones, air-handling units, ducts, and controls. An air-handling unit (AHU) provides the primary conditioning, ventilation, and flow of air and includes cooling and heating coils, dampers, fans, and controls. A single air handler typically serves many zones and several air handlers are utilized in a large commercial building. For each AHU, outdoor ventilation air is mixed with return air from the zones and fed to the cooling coil. Outdoor and return air dampers are typically controlled using an economizer control that selects between minimum and maximum ventilation air depending upon the condition of the outside air. The cooling coil provides both cooling and dehumidification of the process air. The air outlet temperature from

the coil is controlled with a local feedback controller that adjusts the flow of water using a valve. A supply fan and return fan (not shown in Fig. 2) provide the necessary airflow to and from the zones. With a VAV system, zone temperature set points are regulated using a feedback controller applied to dampers within the VAV boxes. The overall air flow provided by the AHU is typically controlled to maintain a duct static pressure set point within the supply duct.

The **chilled water loop** communicates between the cooling coils within the AHUs and chillers that provide the primary source for cooling. It consists of pumps, pipes, valves, and controls. Primary/secondary chilled water systems are commonly employed to accommodate variable-speed pumping. In the primary loop, fixed-speed pumps are used to provide relatively constant chiller flow rates to ensure good performance and reduce the risk of evaporator tube freezing. Individual pumps are typically cycled on and off with a chiller that it serves. The secondary loop incorporates one or more variable-speed pumps that are typically controlled to maintain a set point for chilled water

loop differential pressure between the building supplies and returns.

The primary source of cooling for the system is typically provided by one or more **chillers** that are arranged in parallel and have dedicated pumps. Each chiller has an on-board local-loop feedback controller that adjusts its cooling capacity to maintain a specified set point for chilled water supply temperature. Additional chiller control variables include the number of chillers operating and the relative loading for each chiller. The relative loading can be controlled for a given total cooling requirement by utilizing different chilled water supply set points for constant individual flow or by adjusting individual flows for identical set points. Chillers can be augmented with thermal storage to reduce the amount of chiller power required during occupied periods in order to reduce on-peak energy and power demand costs. The thermal storage medium is cooled during the unoccupied, nighttime period using the chillers when electricity is less expensive. During occupied times, a combination of the chillers and storage are used to meet cooling requirements. Control of thermal storage is defined by the manner in which the storage medium is charged and discharged over time.

The **condenser water loop** includes cooling towers, pumps, piping, and controls. Cooling towers reject energy to the ambient air through heat transfer and possibly evaporation (for wet towers). Larger systems tend to have multiple cooling towers with each tower having multiple cells that share a common sump with individual fans having two or more speed settings. The number of operating cells and tower fan speeds are often controlled using a local-loop feedback controller that maintains a set point for the water temperature leaving the cooling tower. Typically, condenser water pumps are dedicated to individual chillers (i.e., each pump is cycled on and off with a chiller that it serves).

In order to better understand building control variables, interactions, and opportunities, consider how controls change in response to increasing building cooling requirements for the system of Fig. 2. As energy gains to the zones increase, zone temperatures rise in the absence of any control changes. However, zone feedback controllers respond to higher temperatures by increasing VAV box airflow through increased damper openings. This leads to reduced static pressure in the primary supply duct, which causes the AHU supply fan controller to create additional airflow. The greater airflow causes an increase in supply air temperatures leaving the cooling coils in the absence of any additional control changes. However, the supply air temperature feedback controllers respond by opening the cooling coil valves to increase water flow and the heat transfer to the chilled water (the cooling load). For variable-speed pumping, a feedback controller would respond to decreasing pressure differential by increasing the pump speed. The chillers would then experience increased loads due to higher return water temperature and/or flow rate that would lead to increases in chilled water supply temperatures. However, the chiller controllers would respond by increasing chiller cooling capacities in order to maintain the chilled water supply set points (and match the cooling coil loads). In turn, the heat rejection to the condenser water loop would increase to balance the increased energy removed by the chiller, which would increase the temperature of water leaving the condenser. The temperature of water leaving the cooling tower would then increase due to an increase in its energy water temperature. However, a feedback controller would respond to the higher condenser water supply temperature and increase the tower airflow. At some load, the current set of operating chillers would not be sufficient to meet the load (i.e., maintain the chilled water supply set points) and an additional chiller would need to be brought online.

This example illustrated how different local-loop controllers might respond to load changes in order to maintain individual set points. Supervisory control might change these set points and modes of operation. At any given time, it is possible to meet the cooling needs with any number of different modes of operation and set points leading to the potential for control optimization to minimize an objective function.

The system depicted in Fig. 2 and described in the preceding paragraphs represents one of

many different types of systems employed in commercial buildings. Medium-sized commercial buildings often employ multiple direct expansion (DX) cooling systems where refrigerant flows between each AHU and an outdoor condensing unit that employs variable capacity compressors. The compressor capacity is typically controlled to maintain a supply air temperature set point, which is still available as a supervisory control variable. However, the other condensing unit controls (e.g., condensing fans, expansion valve) are typically prepackaged with the unit and not available to the EMCS. For smaller commercial buildings, rooftop units (RTUs) are typically employed that contain a prepackaged AHU, refrigeration cycle, and controls. Each RTU directly cools the air in a portion of the building in response to an individual thermostat. The capacity control is typically on/off staging of the compressor and constant volume air flow is mostly commonly employed. In this case, the only free supervisory control variables are the thermostat set points. In general, the degrees of freedom for supervisory control decrease in going from chilled water cooling plants to DX system to RTUs. In addition, the utility rate incentives for taking advantage of thermal storage and advanced controls are greater for large commercial building applications.

Optimal Supervisory Control

In commercial buildings, it is common to have electric utility rates that have energy and demand charges that vary with time of use. The different rate periods can often include on-peak, off-peak, and mid-peak periods. For this type of rate structure, the time horizon necessary to truly minimize operating costs extends over the entire month. In order to better understand the control issues, consider the general optimal control problem for minimizing monthly electrical utility charges associated with operating an all-electric cooling system in the presence of time-of-use and demand charges. The dynamic optimization involves minimizing

$$J = \sum_{p=1}^{\text{rate periods}} J_p + R_{d,a} \max [P_k]_{k=1 \, to \, N_{\text{month}}}$$

with respect to a trajectory of controls $\vec{u}_k, k = 1 \, to \, N_{\text{month}}$, $\vec{M}_k, k = 1 \, to \, N_{\text{month}}$ where

$$J_p = R_{e,p} \sum_{j=1}^{N_p} P_{p,j} \Delta t + R_{d,p} \max \left[P_{p,j} \right]_{j=1 \, to \, N_p}$$

with the optimization subject to the following general constraints

$$P_k = P(\vec{f}_k, \vec{u}_k, \vec{M}_k)$$
$$\vec{x}_k = x(\vec{x}_{k-1}, \vec{f}_k, \vec{u}_k, \vec{M}_k)$$
$$\vec{u}_{k,\min} \leq \vec{u}_k \leq \vec{u}_{k,\max}$$
$$\mathbf{x}_{k,\min} \leq \mathbf{x}_k \leq \mathbf{x}_{k,\max} \, \vec{y}_k (\vec{f}_k, \vec{u}_k, \vec{M}_k) \leq \vec{y}_{k,\max}$$
$$(\vec{f}_k, \vec{u}_k, \vec{M}_k)$$

where J is the monthly electrical cost (\$), the subscript p denotes that a quantity is limited to a particular type of rate period p (e.g., on-peak, off-peak, mid-peak), $R_{d,a}$ is an anytime demand charge (\$/kW) that is applied to the maximum power consumption occurring over the month P_k is average building power (kW) for stage k within the month, N_{month} is the number of stages in the month, $R_{e,p}$ is the unit cost of electrical energy (\$/kWh) for rate period type p, Δt is the length of the stage (h), N_p is the number of stages within rate period type p in the month, $R_{d,p}$ is a rate period specific demand charge (\$/kW) that is applied to the maximum power consumption occurring during the month within rate period p, $\mathbf{f}_k$ is a vector of uncontrolled inputs that affect building power consumption (e.g., weather, internal gains), $\vec{u}_k$ is a vector of continuous supervisory control variables (e.g., supply air temperature set point), $\mathbf{M}_k$ is a vector of discrete supervisory control variables (chiller on/off controls), $\mathbf{x}_k$ is a vector of state variables, $\mathbf{y}_k$ is a vector of outputs, and subscripts min and max denote minimum and maximum allowable values.

The state variables could characterize the state of a storage device such as a chilled water or

ice storage tank. In this case, the states would be constrained between limits associated with the device's practical storage potential. When variations in zone temperature set points are considered within an optimization, then state variables associated with the distributed nature of energy storage within the building structure are important to consider. The outputs are additional quantities of interest, such as equipment cooling capacities, occupant comfort conditions, etc., that often need to be constrained. In order to implement a model-based predictive control scheme, it would be necessary to have models for the building power, state variables, and outputs in terms of the control and uncontrolled variables. The uncontrolled variables would generally include weather (temperature, humidity, solar radiation) and internal gains due to lights and occupants, etc., that would need to be forecasted over a prediction horizon.

It is not feasible to solve this type of monthly optimization problem for buildings for a variety of reasons, including that forecasting of uncontrolled inputs beyond a day is unreliable. Also, it is very costly to develop the models necessary to implement a model-based control approach of this scale for a particular building. However, it is instructive to consider some special cases that have led to some practical control approaches. First of all, consider the problem of optimizing only the cooling plant supervisory control variables when energy storage effects are not important. This is typically the case for typical systems that do not include ice or chilled water storage. For this scenario, the future does not matter and the problem can be reformulated as a static optimization problem, such that for each stage k the goal is to minimize the building power consumption, $J = P_k$, with respect to the current supervisory control variables, $\vec{u}_k$ and $\mathbf{M}_k$, and subject to constraints. ASHRAE (2011) presents a number of heuristic approaches for adjusting supervisory control variables that have been developed through consideration of this type of optimization problem. This includes algorithms for adjusting cooling tower fan settings, chilled water supply air set points, and chiller sequencing and loading.

Other heuristic approaches have been developed (e.g., ASHRAE 2011; Braun 2007) for controlling the charging and discharging of ice or chilled water storage that were derived from a daily optimization formulation. For the case of real-time pricing of energy, heuristic charging and discharging strategies were derived from minimizing a daily cost function

$$J_{\mathrm{day}} = \sum_{k=1}^{N_{\mathrm{day}}} R_{e,k} P_k \Delta t$$

with respect to a trajectory of charging and discharging rates, subject to a constraint of equal beginning and ending storage states along with other constraints previously described. For the case of typical time-of-use (e.g., on-peak, off-peak) or real-time pricing energy charges with demand charges, heuristic strategies have been developed based on the same form of the daily cost function above with an added demand cost constraint $R_{d,k} P_k \leq TDC$ where TDC is a target demand cost that is set heuristically at the beginning of each billing period and updated at each stage as $TDC_{k+1} = \max(TDC_k, R_{d,k} P_k)$. The heuristic storage control strategies can be readily combined with heuristic strategies for the cooling plant components.

There has been a lot of interest in developing practical methods for dynamic control of zone temperature set points within the bounds of comfort in order to minimize the utility costs. However, this is a very difficult problem and so this remains in the research realm for the time being with limited commercial success.

Summary and Future Directions

Although there is great opportunity for reducing energy use and operating costs in buildings through optimal supervisory control, it is rarely implemented in practice because of high costs associated with engineering site-specific solutions. Current efforts are underway to develop scalable approaches that utilize general methods for configuring and learning models needed to implement model-based predictive control (MPC).

The current thinking is that solutions for optimal supervisory control will be implemented in the cloud and overlay on existing building automation systems (BMS) through the use of universal middleware. This will reduce the cost of implementation compared to programming within existing BMS. There is also a need to reduce the cost of the additional sensors needed to implement MPC. One approach involves the use of virtual sensors that employ models with low-cost sensor inputs to provide higher value information that would normally require expensive sensors to obtain.

Cross-References

Bibliography

ASHRAE (2011) Supervisory control strategies and optimization. In: 2011 ASHRAE handbook of HVAC applications, chap 42. ASHRAE, Atlanta, GA

Braun JE (2007) A near-optimal control strategy for cool storage systems with dynamic electric rates. HVAC&R Res 13(4):557–580

Li H, Yu D, Braun JE (2011) A review of virtual sensing technology and application in building systems. HVAC&R Res 17(5):619–645

Mitchell JW, Braun JE (2013) Principles of heating ventilation and air conditioning in buildings. Wiley, Hoboken

Roth KW, Westphalen D, Feng MY, Llana P, Quartararo L (2005) Energy impact of commercial building controls and performance diagnostics: market characterization, energy impact of building faults, and energy savings potential. TIAX report no. D0180

Wang S, Ma Z (2008) Supervisory and optimal control of building HVAC systems: a review. HVAC&R Res 14(1):3–32

C

CACSD

▶ Computer-Aided Control Systems Design: Introduction and Historical Overview

Cascading Network Failure in Power Grid Blackouts

Ian Dobson
Iowa State University, Ames, IA, USA

Abstract

Cascading failure consists of complicated sequences of dependent failures and can cause large blackouts. The emerging risk analysis, simulation, and modeling of cascading blackouts are briefly surveyed, and key references are suggested.

Keywords

Branching process; Dependent failures; Outage; Power law; Risk; Simulation

Introduction

The main mechanism for the rare and costly widespread blackouts of bulk power transmission systems is cascading failure. Cascading failure can be defined as a sequence of dependent events that successively weaken the power system (IEEE PES CAMS Task Force on Cascading Failure 2008). The events and their dependencies are very varied and include outages or failures of many different parts of the power system and a whole range of possible physical, cyber, and human interactions. The events and dependencies tend to be rare or complicated, since the common and straightforward failures tend to be already mitigated by engineering design or operating practice.

Examples of a small initial outage cascading into a complicated sequence of dependent outages are the August 10, 1996, blackout of the Northwest United States that disconnected power to about 7.5 million customers (Kosterev et al. 1999) and the August 14, 2003 blackout of about 50 million customers in Northeastern United States and Canada (US-Canada Power System Outage Task Force 2004). Although such extreme events are rare, the direct costs run to billions of dollars and the disruption to society is substantial. Large blackouts also have a strong effect on shaping the way power systems are regulated and the reputation of the power industry. Moreover, some blackouts involve social disruptions that can multiply the economic damage. The hardship to people and possible deaths underscore the engineer's responsibility to work to avoid blackouts.

It is useful when analyzing cascading failure to consider cascading events of all sizes, including the short cascades that do not lead to interruption

of power to customers and cascades that involve events in other infrastructures, especially since loss of electricity can significantly impair other essential or economically important infrastructures. Note that in the context of interacting infrastructures, the term "cascading failure" sometimes has the more restrictive definition of events cascading *between* infrastructures (Rinaldi et al. 2001).

Blackout Risk

Cascading failure is a sequence of dependent events that successively weaken the power system. At a given stage in the cascade, the previous events have weakened the power system so that further events are more likely. It is this dependence that makes the long series of cascading events that cause large blackouts likely enough to pose a substantial risk. (If the events were independent, then the probability of a large number of events would be the product of the small probabilities of individual events and would be vanishingly small.) The statistics for the distribution of sizes of blackouts have correspondingly "heavy tails" indicating that blackouts of all sizes, including large blackouts, can occur. Large blackouts are rare, but they are expected to happen occasionally, and they are not "perfect storms."

In particular, it has been observed in several developed countries that the probability distribution of blackout size has an approximate power law dependence (Carreras et al. 2004b; Dobson et al. 2007; Hines et al. 2009). (The power law is of course limited in extent because every grid has a largest possible blackout in which the entire grid blacks out.) The power law region can be explained using ideas from complex systems theory. The main idea is that over the long term, the power grid reliability is shaped by the engineering responses to blackouts and the slow load growth and tends to evolve towards the power law distribution of blackout size (Dobson et al. 2007; Ren et al. 2008).

Blackout risk can be defined as the product of blackout probability and blackout cost. One simple assumption is that blackout cost is roughly proportional to blackout size, although larger blackouts may well have costs (especially indirect costs) that increase faster than linearly. In the case of the power law dependence, the larger blackouts can become rarer at a similar rate as costs increase, and then the risk of large blackouts is comparable to or even exceeding the risk of small blackouts. Mitigation of blackout risk should consider both small and large blackouts, because mitigating the small blackouts that are easiest to analyze may inadvertently increase the risk of large blackouts (Newman et al. 2011).

Approaches to quantify blackout risk are challenging and emerging, but there are also valuable approaches to mitigating blackout risk that do not quantify the blackout risk. The n-1 criterion that requires the power system to survive any single component failure has the effect of reducing cascading failures. The individual mechanisms of dependence in cascades (overloads, protection failures, voltage collapse, transient stability, lack of situational awareness, human error, etc.) can be addressed individually by specialized analyses or simulations or by training and procedures. Credible initiating outages can be sampled and simulated, and those resulting in cascading can be mitigated (Hardiman et al. 2004). This can be thought of as a "defense in depth" approach in which mitigating a subset of credible contingencies is likely to mitigate other possible contingencies not studied. Moreover, when blackouts occur, a postmortem analysis of that particular sequence of events leads to lessons learned that can be implemented to mitigate the risk of some similar blackouts (US-Canada Power System Outage Task Force 2004).

Simulation and Models

There are many simulations of cascading blackouts using Monte Carlo and other methods, for example, Hardiman et al. (2004), Carreras et al. (2004a), Chen et al. (2005), Kirschen et al. (2004), Anghel et al. (2007), and Bienstock and Mattia (2007). All these simulations select and approximate a modest subset of the many physical and engineering mechanisms of

cascading failure, such as line overloads, voltage collapse, and protection failures. In addition, operator actions or the effects of engineering the network may also be crudely represented. Some of the simulations give a set of credible cascades, and others approximately estimate blackout risk.

Except for describing the initial outages, where there is a wealth of useful knowledge, much of standard risk analysis and modeling does not easily apply to cascading failure in power systems because of the variety of dependencies and mechanisms, the combinatorial explosion of rare possibilities, and the heavy-tailed probability distributions. However, progress has been made in probabilistic models of cascading (Chen et al. 2006; Dobson 2012; Rahnamay-Naeini et al. 2012).

The goal of high-level probabilistic models is to capture salient features of the cascade without detailed models of the interactions and dependencies. They provide insight into cascading failure data and simulations, and parameters of the high-level models can serve as metrics of cascading.

Branching process models are transient Markov probabilistic models in which, after some initial outages, the outages are produced in successive generations. Each outage in each generation (a "parent" outage) produces a probabilistic number of outages ("children" outages) in the next generation according to an offspring probability distribution. The children failures then become parents to produce the next generation and so on, until there is a generation with zero children and the cascade stops. As might be expected, a key parameter describing the cascading is its average propagation, which is the average number of children outages per parent outage. Branching processes have traditionally been applied to many cascading processes outside of risk analysis (Harris), but they have recently been validated and applied to estimate the distribution of the total number of outages from utility outage data (Dobson 2012). A probabilistic model that tracks the cascade as it progresses in time through lumped grid states is presented in Rahnamay-Naeini et al. (2012).

There is an extensive complex networks literature on cascading in abstract networks that is largely motivated by idealized models of propagation of failures in the Internet. The way that failures propagate only along the network links is not realistic for power systems, which satisfy Kirchhoff's laws so that many types of failures propagate differently. For example, line overloads tend to propagate along cutsets of the network. However, the high-level qualitative results of phase transitions in the complex networks have provided inspiration for similar effects to be discovered in power system models (Dobson et al. 2007). There is also a possible research opportunity to elaborate the complex network models to incorporate some of the realities of power system and then validate them.

Summary and Future Directions

One challenge for simulation is what selection of phenomena to model and in how much detail in order to get useful engineering results. Faster simulations would help to ease the requirements of sampling appropriately from all the sources of uncertainty. Better metrics of cascading in addition to average propagation need to be developed and extracted from real and simulated data in order to better quantify and understand blackout risk. There are many new ideas emerging to analyze and simulate cascading failure, and the next step is to validate and improve these new approaches by comparing them with observed blackout data. Overall, there is an exciting challenge to build on the more deterministic approaches to mitigate cascading failure and find ways to more directly quantify and mitigate cascading blackout risk by coordinated analysis of real data, simulation, and probabilistic models.

Cross-References

▶ Hybrid Dynamical Systems, Feedback Control of
▶ Lyapunov Methods in Power System Stability
▶ Power System Voltage Stability
▶ Small Signal Stability in Electric Power Systems

Bibliography

Anghel M, Werley KA, Motter AE (2007) Stochastic model for power grid dynamics. In: 40th Hawaii international conference on system sciences, Hawaii, Jan 2007

Bienstock D, Mattia S (2007) Using mixed-integer programming to solve power grid blackout problems. Discret Optim 4(1):115–141

Carreras BA, Lynch VE, Dobson I, Newman DE (2004a) Complex dynamics of blackouts in power transmission systems. Chaos 14(3):643–652

Carreras BA, Newman DE, Dobson I, Poole AB (2004b) Evidence for self-organized criticality in a time series of electric power system blackouts. IEEE Trans Circuits Syst Part I 51(9):1733–1740

Chen J, Thorp JS, Dobson I (2005) Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model. Int J Electr Power Energy Syst 27(4):318–326

Chen Q, Jiang C, Qiu W, McCalley JD (2006) Probability models for estimating the probabilities of cascading outages in high-voltage transmission network. IEEE Trans Power Syst 21(3): 1423–1431

Dobson I, Carreras BA, Newman DE (2005) A loading-dependent model of probabilistic cascading failure. Probab Eng Inf Sci 19(1):15–32

Dobson I, Carreras BA, Lynch VE, Newman DE (2007) Complex systems analysis of series of blackouts: cascading failure, critical points, and self-organization. Chaos 17:026103

Dobson I (2012) Estimating the propagation and extent of cascading line outages from utility data with a branching process, IEEE Trans Power Systems 27(4): 2146–215

Hardiman RC, Kumbale MT, Makarov YV (2004) An advanced tool for analyzing multiple cascading failures. In: Eighth international conference on probability methods applied to power systems, Ames, Sept 2004

Harris TE (1989) Theory of branching processes. Dover, New York

Hines P, Apt J, Talukdar S (2009) Large blackouts in North America: historical trends and policy implications. Energy Policy 37(12):5249–5259

IEEE PES CAMS Task Force on Cascading Failure (2008) Initial review of methods for cascading failure analysis in electric power transmission systems. In: IEEE power and energy society general meeting, Pittsburgh, July 2008

Kirschen DS, Strbac G (2004) Why investments do not prevent blackouts. Electr J 17(2):29–36

Kirschen DS, Jawayeera D, Nedic DP, Allan RN (2004) A probabilistic indicator of system stress. IEEE Trans Power Syst 19(3):1650–1657

Kosterev D, Taylor C, Mittelstadt W (1999) Model validation for the August 10, 1996 WSCC system outage. IEEE Trans Power Syst 14:967–979

Newman DE, Carreras BA, Lynch VE, Dobson I (2011) Exploring complex systems aspects of blackout risk and mitigation. IEEE Trans Reliab 60(1): 134–143

Rahnamay-Naeini M, Wang Z, Ghani N, Mammoli A, Hayat M.M (2014) Stochastic Analysis of Cascading-Failure Dynamics in Power Grids, to appear in IEEE Transactions on Power Systems

Ren H, Dobson I, Carreras BA (2008) Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid. IEEE Trans Power Syst 23(3):1217–1225

Rinaldi SM, Peerenboom JP, Kelly TK (2001) Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Syst Mag 21:11–25

US-Canada Power System Outage Task Force (2004) Final report on the August 14, 2003 blackout in the United States and Canada

Cash Management

Abel Cadenillas
University of Alberta, Edmonton, AB, Canada

Abstract

Cash on hand (or cash held in highly liquid form in a bank account) is needed for routine business and personal transactions. The problem of determining the right amount of cash to hold involves balancing liquidity against investment opportunity costs. This entry traces solutions using both discrete-time and continuous-time stochastic models.

Keywords

Brownian motion; Inventory theory; Stochastic impulse control

Definition

A firm needs to keep cash, either in the form of cash on hand or as a bank deposit, to meet its daily transaction requirements. Daily inflows

and outflows of cash are random. There is a finite target for the cash balance, which could be zero in some cases. The firm wants to select a policy that minimizes the expected total discounted cost for being far away from the target during some time horizon. This time horizon is usually infinity. The firm has an incentive to keep the cash level low, because each unit of positive cash leads to a holding cost since cash has alternative uses like dividends or investments in earning assets. The firm has an incentive to keep the cash level high, because penalty costs are generated as a result of delays in meeting demands for cash. The firm can increase its cash balance by raising new capital or by selling some earnings assets, and it can reduce its cash balance by paying dividends or investing in earning assets. This control of the cash balance generates fixed and proportional transaction costs. Thus, there is a cost when the cash balance is different from its target, and there is also a cost for increasing or reducing the cash reserve. The objective of the manager is to minimize the expected total discounted cost.

Hasbrouck (2007), Madhavan and Smidt (1993), and Manaster and Mann (1996) study inventories of stocks that are similar to the cash management problem.

The Solution

The qualitative form of optimal policies of the cash management problem in discrete time was studied by Eppen and Fama (1968, 1969), Girgis (1968), and Neave (1970). However, their solutions were incomplete.

Many of the difficulties that they and other researchers encountered in a discrete-time framework disappeared when it was assumed that decisions were made continuously in time and that demand is generated by a Brownian motion with drift. Vial (1972) formulated the cash management problem in continuous time with fixed and proportional transaction costs, linear holding and penalty costs, and demand for cash generated by a Brownian motion with drift. Under very

strong assumptions, Vial (1972) proved that if an optimal policy exists, then it is of a simple form (a, α, β, b).

This means that the cash balance should be increased to level α when it reaches level a and should be reduced to level β when it reaches level b. Constantinides (1976) assumed that an optimal policy exists and it is of a simple form, and determined the above levels and discussed the properties of the optimal solution. Constantinides and Richard (1978) proved the main assumptions of Vial (1972) and therefore obtained rigorously a solution for the cash management problem.

Constantinides and Richard (1978) applied the theory of stochastic impulse control developed by Bensoussan and Lions (1973, 1975, 1982). He used a Brownian motion W to model the uncertainty in the inventory. Formally, he considered a probability space $(\Omega, \mathcal{F}, P)$ together with a filtration $(\mathcal{F}_t)$ generated by a one-dimensional Brownian motion W. He considered $X_t :=$ inventory level at time t, and assumed that X is an adapted stochastic process given by

$$X_t = x - \int_0^t \mu \, ds - \int_0^t \sigma \, dW_s + \sum_{i=1}^{\infty} I_{\{\tau_i < t\}} \xi_i.$$

Here, $\mu > 0$ is the drift of the demand and $\sigma > 0$ is the volatility of the demand. Furthermore, τ_i is the time of the i-th intervention and ξ_i is the intensity of the i-th intervention.

A stochastic impulse control is a pair

$$((\tau_n); (\xi_n))$$
$$= (\tau_0, \tau_1, \tau_2, \ldots, \tau_n, \ldots; \xi_0, \xi_1, \xi_2, \ldots, \xi_n, \ldots),$$

where

$$\tau_0 = 0 < \tau_1 < \tau_2 < \cdots < \tau_n < \cdots$$

is an increasing sequence of stopping times and (ξ_n) is a sequence of random variables such that each $\xi_n : \Omega \mapsto \mathbf{R}$ is measurable with respect

to $\mathcal{F}_{\tau_n}$. We assume $\xi_0 = 0$. The management (the controller) decides to act at time

$$X_{\tau_i^+} = X_{\tau_i} + \xi_i.$$

We note that ξ_i and X can also take negative values. The management wants to select the pair

$$((\tau_n); (\xi_n))$$

that minimizes the functional J defined by

$$J(x; ((\tau_n); (\xi_n))) := E\left[\int_0^\infty e^{-\lambda t} f(X_t)dt \right.$$
$$\left. + \sum_{n=1}^\infty e^{-\lambda \tau_n} g(\xi_n) I_{\{\tau_n < \infty\}}\right],$$

where

$$f(x) = \max(hx, -px)$$

and

$$g(\xi) = \begin{cases} C + c\xi & \text{if } \xi > 0 \\ \min(C, D) & \text{if } \xi = 0 \\ D - d\xi & \text{if } \xi < 0 \end{cases}$$

Furthermore, $\lambda > 0, C, c, D, d \in (0, \infty)$, and $h, p \in (0, \infty)$. Here, f represents the running cost incurred by deviating from the aimed cash level 0, C represents the fixed cost per intervention when the management pushes the cash level upwards, D represents the fixed cost per intervention when the management pushes the cash level downwards, c represents the proportional cost per intervention when the management pushes the cash level upwards, d represents the proportional cost per intervention when the management pushes the cash level downwards, and λ is the discount rate.

The results of Constantinides were complemented, extended, or improved by Cadenillas et al. (2010), Cadenillas and Zapatero (1999), Feng and Muthuraman (2010), Harrison et al. (1983), and Ormeci et al. (2008).

Cross-References

▶ Financial Markets Modeling
▶ Inventory Theory

Bibliography

Bensoussan A, Lions JL (1973) Nouvelle formulation de problemes de controle impulsionnel et applications. C R Acad Sci (Paris) Ser A 276:1189–1192

Bensoussan A, Lions JL (1975) Nouvelles methodes en controle impulsionnel. Appl Math Opt 1:289–312

Bensoussan A, Lions JL (1982) Controle impulsionnel et inequations quasi variationelles. Bordas, Paris

Cadenillas A, Zapatero F (1999) Optimal Central Bank intervention in the foreign exchange market. J Econ Theory 87:218–242

Cadenillas A, Lakner P, Pinedo M (2010) Optimal control of a mean-reverting inventory. Oper Res 58:1697–1710

Constantinides GM (1976) Stochastic cash management with fixed and proportional transaction costs. Manage Sci 22:1320–1331

Constantinides GM, Richard SF (1978) Existence of optimal simple policies for discounted-cost inventory and cash management in continuous time. Oper Res 26:620–636

Eppen GD, Fama EF (1968) Solutions for cash balance and simple dynamic portfolio problems. J Bus 41:94–112

Eppen GD, Fama EF (1969) Cash balance and simple dynamic portfolio problems with proportional costs. Int Econ Rev 10:119–133

Feng H, Muthuraman K (2010) A computational method for stochastic impulse control problems. Math Oper Res 35:830–850

Girgis NM (1968) Optimal cash balance level. Manage Sci 15:130–140

Harrison JM, Sellke TM, Taylor AJ (1983) Impulse control of Brownian motion. Math Oper Res 8:454–466

Hasbrouck J (2007) Empirical market microstructure. Oxford University Press, New York

Madhavan A, Smidt S (1993) An analysis of changes in specialist inventories and quotations. J Finance 48:1595–1628

Manaster S, Mann SC (1996) Life in the pits: competitive market making and inventory control. Rev Financ Stud 9:953–975

Neave EH (1970) The stochastic cash-balance problem with fixed costs for increases and decreases. Manage Sci 16:472–490

Ormeci M, Dai JG, Vande Vate J (2008) Impulse control of Brownian motion: the constrained average cost case. Oper Res 56:618–629

Vial JP (1972) A continuous time model for the cash balance problem. In: Szego GP, Shell C (eds) Mathematical methods in investment and finance. North Holland, Amsterdam

Classical Frequency-Domain Design Methods

J. David Powell and Abbas Emami-Naeini
Stanford University, Stanford, CA, USA

Abstract

The design of feedback control systems in industry is probably accomplished using frequency-response (FR) methods more often than any other. Frequency-response design is popular primarily because it provides good designs in the face of uncertainty in the plant model ($G(s)$ in Fig. 1). For example, for systems with poorly known or changing high-frequency resonances, we can temper the feedback design to alleviate the effects of those uncertainties. Currently, this tempering is carried out more easily using FR design than with any other method. The method is most effective for systems that are stable in open loop; however, it can also be applied to systems with instabilities. This section will introduce the reader to methods of design (i.e., finding $D(s)$ in Fig. 1) using lead and lag compensation. It will also cover the use of FR design to reduce steady-state errors and to improve robustness to uncertainties in high-frequency dynamics.

Keywords

Amplitude stabilization; Bandwidth; Bode plot; Crossover frequency; Frequency response; Gain; Gain stabilization; Gain margin; Notch filter; Phase; Phase margin; Stability

Introduction

Finding an appropriate compensation ($D(s)$ in Fig. 1) using frequency response is probably the easiest of all feedback control design methods. Designs are achievable starting with the FR plots of both magnitude and phase of $G(s)$ then selecting $D(s)$ to achieve certain values of the

gain and/or phase margins and system bandwidth or error characteristics. This section will cover the design process for finding an appropriate $D(s)$.

Design Specifications

As discussed in Section X, the **gain margin (GM)** is the factor by which the gain can be raised before instability results. The **phase margin (PM)** is the amount by which the phase of $D(j\omega)G(j\omega)$ exceeds $-180°$ when $|D(j\omega)G(j\omega)| = 1$, the **crossover frequency**. Performance requirements for control systems are often partially specified in terms of PM and/or GM. For example, a typical specification might include the requirement that PM > 50° and GM > 5. It can be shown that the PM tends to correlate well with the damping ratio, ζ, of the closed-loop roots. In fact, it is shown in Franklin et al. (2010), that

$$\zeta \cong \frac{\text{PM}}{100}$$

for many systems; however, the actual resulting damping and/or response overshoot of the final closed-loop system will need to be verified if they are specified as well as the PM. A PM of 50° would tend to yield a ζ of 0.5 for the closed-loop roots, which is a modestly damped system. The GM does not generally correlate directly with the damping ratio, but is a measure of the degree of stability and is a useful secondary specification to ensure stability.

Another design specification is the **bandwidth**, which was defined in Section X. The bandwidth is a direct measure of the frequency at which the closed-loop system starts to fail in following the input command. It is also a measure of the speed of response of a closed-loop system. Generally speaking, it correlates well with the step response rise time of the system.

In some cases, the **steady-state error** must be less than a certain amount. As discussed in Franklin et al. (2010), the steady-state error is a direct function of the low-frequency gain of
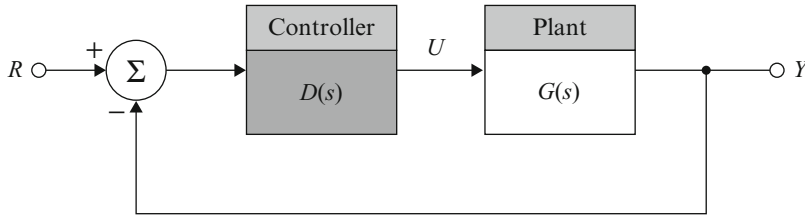
Classical Frequency-Domain Design Methods, Fig. 1 Feedback system showing compensation, $D(s)$ (Source: Franklin et al. (2010, p-249), Reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

the FR magnitude plot. However, increasing the low-frequency gain typically will raise the entire magnitude plot upward, thus increasing the magnitude 1 crossover frequency and, therefore, increasing the speed of response and bandwidth of the system.

Compensation Design

In some cases, the design of a feedback compensation can be accomplished by using proportional control only, i.e., setting $D(s) = K$ (see Fig. 1) and selecting a suitable value for K. This can be accomplished by plotting the magnitude and phase of $G(s)$, looking at $|G(j\omega)|$ at the frequency where $\angle G(j\omega) = -180°$, and then selecting K so that $|KG(j\omega)|$ yields the desired GM. Similarly, if a particular value of PM is desired, one can find the frequency where $\angle G(j\omega) = -180° +$ the desired PM. The value of $|KG(j\omega)|$ at that frequency must equal 1; therefore, the value of $|G(j\omega)|$ must equal $1/K$. Note that the $|KG(j\omega)|$ curve moves vertically based on the value of K; however the $\angle KG(j\omega)$ curve is not affected by the value of K. This characteristic simplifies the design process.

In more typical cases, proportional feedback alone is not sufficient. There is a need for a certain damping (i.e., PM) and/or speed of response (i.e., bandwidth) and there is no value of K that will meet the specifications. Therefore, some increased damping from the compensation is required. Likewise, a certain steady-state error requirement and its resulting low-frequency gain will cause the $|D(j\omega)G(j\omega)|$ to be greater than desired for an acceptable PM, so more phase lead is required from the compensation. This is

typically accomplished by **lead compensation**. A phase increase (or lead) is accomplished by placing a zero in $D(s)$. However, that alone would cause an undesirable high-frequency gain which would amplify noise; therefore, a first-order pole is added in the denominator at frequencies substantially higher than the zero break point of the compensator. Thus, the phase lead still occurs, but the amplification at high frequencies is limited. The resulting lead compensation has a transfer function of
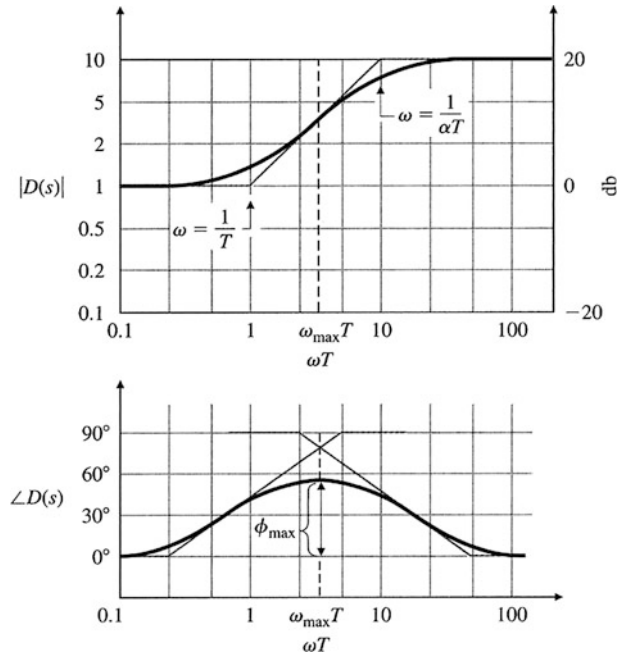
$$D(s) = K\frac{Ts + 1}{\alpha Ts + 1}, \qquad \alpha < 1, \qquad (1)$$

where $1/\alpha$ is the ratio between the pole/zero break-point frequencies. Figure 2 shows the frequency response of this lead compensation. The maximum amount of phase lead supplied is dependent on the ratio of the pole to zero and is shown in Fig. 3 as a function of that ratio.

For example, a lead compensator with a zero at $s = -2$ ($T = 0.5$) and a pole at $s = -10$ ($\alpha T = 0.1$) (and thus $\alpha = \frac{1}{5}$) would yield the maximum phase lead of $\phi_{max} = 40°$. Note from the figure that we could increase the phase lead almost up to $90°$ using higher values of the **lead ratio**, $1/\alpha$; however, Fig. 2 shows that increasing values of $1/\alpha$ also produces higher amplifications at higher frequencies. Thus, our task is to select a value of $1/\alpha$ that is a good compromise between an acceptable PM and acceptable noise sensitivity at high frequencies. Usually the compromise suggests that a lead compensation should contribute a maximum of $70°$ to the phase. If a greater phase lead is needed, then a double-lead compensation would be suggested, where

Classical Frequency-Domain Design Methods, Fig. 2 Lead-compensation frequency response with $1/\alpha = 10$, $K = 1$ (Source: Franklin et al. (2010, p-349), Reprinted by permission of Pearson Education, Inc.)

$$D(s) = K\left(\frac{Ts+1}{\alpha Ts+1}\right)^2.$$

Even if a system had negligible amounts of noise present, the pole must exist at some point because of the impossibility of building a pure differentiator. No physical system – mechanical or electrical or digital – responds with infinite amplitude at infinite frequencies, so there will be a limit in the frequency range (or bandwidth) for which derivative information (or phase lead) can be provided.

As an example of designing a lead compensator, let us design compensation for a DC motor with the transfer function

$$G(s) = \frac{1}{s(s+1)}.$$

We wish to obtain a steady-state error of less than 0.1 for a unit-ramp input and we desire a system bandwidth greater than 3 rad/sec. Furthermore, we desire a PM of 45°. To accomplish the error requirement, Franklin et al. shows that

$$e_{ss} = \lim_{s \to 0} s\left[\frac{1}{1 + D(s)G(s)}\right]R(s), \quad (2)$$

and if $R(s) = 1/s^2$ for a unit ramp, Eq. (2) reduces to

$$e_{ss} = \lim_{s \to 0}\left\{\frac{1}{s + D(s)[1/(s+1)]}\right\} = \frac{1}{D(0)}.$$

Therefore, we find that $D(0)$, the steady-state gain of the compensation, cannot be less than 10 if it is to meet the error criterion, so we pick $K = 10$. The frequency response of $KG(s)$ in Fig. 4 shows that the PM $= 20°$ if no phase lead is added by compensation. If it were possible to simply add phase without affecting the magnitude, we would need an additional phase of only 25° at the $KG(s)$ crossover frequency of $\omega = 3$ rad/sec. However, maintaining the same low-frequency gain and adding a compensator zero will increase the crossover frequency; hence, more than a 25° phase contribution will be required from the lead compensation. To be safe, we will design the lead compensator so that it supplies a maximum phase lead of 40°. Figure 3 shows that $1/\alpha = 5$ will accomplish that goal. We will derive the greatest benefit from the compensation if the maximum phase lead from the compensator occurs at the crossover frequency. With some trial and error, we determine

Classical Frequency-Domain Design Methods, Fig. 4 Frequency response for lead-compensation design (Source:
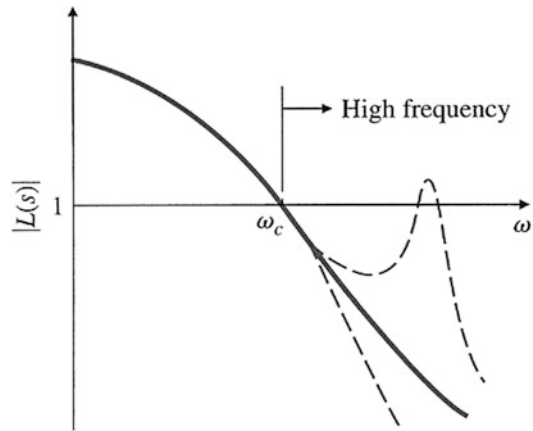Franklin et al. (2010, p-352), Reprinted by permission of Pearson Education, Inc.)

that placing the zero at $\omega = 2$ rad/sec and the pole at $\omega = 10$ rad/sec causes the maximum phase lead to be at the crossover frequency. The compensation, therefore, is

$$D(s) = 10 \frac{s/2 + 1}{s/10 + 1}.$$

The frequency-response characteristics of $L(s) = D(s)G(s)$ in Fig. 4 can be seen to yield a PM of 53°, which satisfies the PM and steady-state error design goals. In addition, the crossover frequency of 5 rad/sec will also yield a bandwidth greater than 3 rad/sec, as desired.

Lag compensation is the same form as the lead compensation in Eq. (1) except that $\alpha > 1$. Therefore, the pole is at a lower frequency than the zero and it produces higher gain at lower frequencies. The compensation is used primarily to reduce steady-state errors by raising the low-frequency gain but without increasing the crossover frequency and speed of response. This can be accomplished by placing the pole and zero of the lag compensation well below the crossover frequency. Alternatively, lag compensation can also be used to improve the PM by keeping the low frequency gain the same, but reducing the gain near crossover, thus reducing the crossover frequency. That will usually improve the PM since the phase of the uncompensated system typically is higher at lower frequencies.

Systems being controlled often have high-frequency dynamic phenomena, such as mechanical resonances, that could have an impact on the stability of a system. In very-high-performance designs, these high-frequency dynamics are included in the plant model, and a compensator is designed with a specific knowledge of those dynamics. However, a more **robust** approach for designing with uncertain high-frequency dynamics is to keep the high-frequency gain low, just as we did for sensor-noise reduction. The reason for this can be seen from the gain–frequency relationship of a typical system, shown in Fig. 5. The only way instability can result from high-frequency dynamics is if an unknown high-frequency resonance causes the magnitude to rise above 1.

Classical Frequency-Domain Design Methods, Fig. 5 Effect of high-frequency plant uncertainty (Source: Franklin et al. (2010, p-372), Reprinted by permission of Pearson Education, Inc.)

Conversely, if all unknown high-frequency phenomena are guaranteed to remain below a magnitude of 1, stability can be guaranteed. The likelihood of an unknown resonance in the plant G rising above 1 can be reduced if the nominal high-frequency loop gain (L) is lowered by the addition of extra poles in $D(s)$. When the stability of a system with resonances is assured by tailoring the high-frequency magnitude never to exceed 1, we refer to this process as **amplitude** or **gain stabilization**. Of course, if the resonance characteristics are known exactly and remain the same under all conditions, a specially tailored compensation, such as a **notch filter** at the resonant frequency, can be used to tailor the phase for stability even though the amplitude does exceed magnitude 1 as explained in Franklin et al. (2010). Design of a notch filter is more easily carried out using **root locus** or **state-space** design methods, all of which are discussed in Franklin et al. (2010). This method of stabilization is referred to as **phase stabilization**. A drawback to phase stabilization is that the resonance information is often not available with adequate precision or varies with time; therefore, the method is more susceptible to errors in the plant model used in the design. Thus, we see that sensitivity to plant uncertainty

and sensor noise are both reduced by sufficiently low gain at high-frequency.

Summary and Future Directions

Before the common use of computers in design, frequency-response design was the only widely used method. While it is still the most widely used method for routine designs, complex systems and their design are being carried out using a multitude of methods. This section introduces just one of many possible methods.

Cross-References

Bibliography

Franklin GF, Powell JD, Workman M (1998) Digital control of dynamic systems, 3rd edn. Ellis-Kagle Press, Half Moon Bay
Franklin GF, Powell JD, Emami-Naeini A (2010) Feedback control of dynamic systems, 6th edn. Pearson, Upper Saddle River
Franklin GF, Powell JD, Emami-Naeini A (2015) Feedback control of dynamic systems, 7th edn. Pearson, Upper Saddle River

Computational Complexity Issues in Robust Control

Onur Toker
Fatih University, Istanbul, Turkey

Abstract

Robust control theory has introduced several new and challenging problems for researchers. Some of these problems have been solved by innovative approaches and led to the development of new and efficient algorithms. However, some of the other problems in robust control theory had attracted significant amount of research, but none of the proposed algorithms were efficient, namely, had execution time bounded by a polynomial of the "problem size." Several important problems in robust control theory are either of decision type or of computation/approximation type, and one would like to have an algorithm which can be used to answer all or most of the possible cases and can be executed on a classical computer in reasonable amount of time. There is a branch of theoretical computer science, called theory of computation, which can be used to study the difficulty of problems in robust control theory. In the following, classical computer system, algorithm, efficient algorithm, unsolvability, tractability, *NP*-hardness, and *NP*-completeness will be introduced in a more rigorous fashion, with applications to problems from robust control theory.

Keywords

Approximation complexity; Computational complexity; *NP*-complete; *NP*-hard; Unsolvability

Introduction

The term algorithm is used to refer to **different** notions which are all somewhat consistent with our intuitive understanding. This ambiguity may sometimes generate significant confusion, and therefore, a rigorous definition is of extreme importance. One commonly accepted "intuitive" definition is a set of rules that a person can perform with paper and pencil. However, there are "algorithms" which involve random number generation, for example, finding a primitive root in $\mathbb{Z}_p$ (Knuth 1997). Based on this observation, one may ask whether a random number generation-based set of rules should be also considered as an algorithm, provided that it will terminate after finitely many steps for all instances of the

problem or for a significant majority of the cases. In a similar fashion, one may ask whether any real number, including irrational ones which cannot be represented on a digital computer without an approximation error, should be allowed as an input to an algorithm and, furthermore, should all calculations be limited to algebraic functions only or should exact calculation of non-algebraic functions, e.g., trigonometric functions, the gamma function, etc., be acceptable in an algorithm. Although all of these seem acceptable with respect to our intuitive understanding of the algorithm, from a rigorous point of view, they are different notions. In the context of robust control theory, as well as many other engineering disciplines, there is a separate and widely accepted definition of algorithm, which is based on today's digital computers, more precisely the Turing machine (Turing 1936). Alan M. Turing defined a "hypothetical computation machine" to formally define the notions of algorithm and computability. A Turing machine is, in principle, quite similar to today's digital computers widely used in many engineering applications. The engineering community seems to widely accept the use of current digital computers and Turing's definitions of algorithm and computability.

However, depending on new scientific, engineering, and technological developments, superior computation machines may be constructed. For example, there is no guarantee that quantum computing research will not lead to superior computation machines (Chen et al. 2006; Kaye et al. 2007). In the future, the engineering community may feel the need to revise formal definitions of algorithm, computability, tractability, etc., if such superior computation machines can be constructed and used for scientific/engineering applications.

Turing Machines and Unsolvability
Turing machine is basically a mathematical model of a simplified computation device. The original definition involves a tape-like device for memory. For an easy-to-read introduction to the Turing machine model, see Garey and Johnson (1979) and Papadimitriou (1995), and for more details, see Hopcroft et al. (2001),

Lewis and Papadimitriou (1998), and Sipser (2006). Despite this being a quite simple and low-performance "hardware" compared to today's engineering standards, the following two observations justify their use in the study of computational complexity of engineering problems. Anything which can be solved on today's current digital computers can be solved on a Turing machine. Furthermore, a polynomial-time algorithm on today's digital computers will correspond to again a polynomial-time algorithm on the original Turing machine, and vice versa. A widely accepted definition for an algorithm is a Turing machine with a program, which is guaranteed to terminate after finitely many steps.

For some mathematical and engineering problems, it can be shown that there can be no algorithm which can handle all possible cases. Such problems are called unsolvable. The condition "all cases" may be considered too tough, and one may argue that such negative results have only theoretical importance and have no practical implications. But such results do imply that we should concentrate our efforts on alternative research directions, like the development of algorithms only for cases which appear more frequently in real scientific/engineering applications, without asking the algorithm to work for the remaining cases as well.

Here is a famous unsolvable mathematical problem: Hilbert's tenth problem is basically the development of an algorithm for testing whether a Diophantine equation has an integer solution. However, in 1970, Matijasevich showed that there can be no such algorithm (Matiyasevich 1993). Therefore, we say that the problem of checking whether a Diophantine equation has an integer solution is unsolvable.

Several unsolvability results for dynamical systems can be proved by using the Post correspondence problem (Davis 1985) and the embedding of free semigroups into matrices. For example, the problem of checking the stability of saturated linear dynamical systems is proved to be undecidable (Blondel et al. 2001), meaning that no general stability test algorithm can be developed for such systems. A similar unsolvability result is reported in Blondel and

Tsitsiklis (2000a) for boundedness of switching systems of the type

$$x(k + 1) = A_{f(k)}x(k),$$

where f is assumed to be an arbitrary and unknown function from $\mathbb{N}$ into $\{0, 1\}$. A closely related asymptotic stability problem is equivalent to testing whether the joint spectral radius (JSR) (Rota and Strang 1960) of a set of matrices is less than one. For a quite long period of time, there was a conjecture called the finiteness conjecture (FC) (Lagarias and Wang 1995), which was generally believed or hoped to be true, at least for a group of researchers. FC may be interpreted as "For asymptotic stability of $x(k + 1) = A_{f(k)}x(k)$ type switching systems, it is enough to consider periodic switchings only." There was no known counterexample, and the truth of this conjecture would imply existence of an algorithm for the abovementioned JSR problem. However, it was shown in Bousch and Mairesse (2002) that FC is not true (see Blondel et al. (2003) for a simplified proof). There are numerous known computationally very valuable procedures related to JSR approximation, for example, see Blondel and Nesterov (2005) and references therein. However, the development of an algorithm to test whether JSR is less than one remains as an open problem.

For further results on unsolvability and unsolved problems in robust control, see Blondel et al. (1999), Blondel and Megretski (2004), and references therein.

Tractability, *NP*-Hardness, and *NP*-Completeness

The engineering community is interested in not only solution algorithms but algorithms which are fast even in the worst case and if not on the average. Sometimes, this speed requirement may be relaxed to being fast for most of the cases and sometimes to only a significant percentage of the cases. Currently, the theory of computation is developed around the idea of algorithms which are polynomial time even in the worst case, namely, execution time bounded by a polynomial of the problem size (Garey and Johnson 1979;

Papadimitriou 1995). Such algorithms are also called efficient, and associated problems are classified as tractable. The term problem size means number of bits used in a suitable encoding of the problem (Garey and Johnson 1979; Papadimitriou 1995).

One may argue that this worst-case approach of being always polynomial time is a quite conservative requirement. In reality, a practicing engineer may consider being polynomial time on the average quite satisfactory for many applications. The same may be true for algorithms which are polynomial time for most of the cases. However, the existing computational complexity theory is developed around this idea of being polynomial time even in the worst case. Therefore, many of the computational complexity results proved in the literature do not imply the impossibility of algorithms which are neither polynomial time on the average nor polynomial time for most of the cases. Note that despite not being efficient, such algorithms may be considered quite valuable by a practicing engineer. Tractability and efficiency can be defined in several different ways, but the abovementioned polynomial-time solvability even in the worst-case approach is widely adopted by the engineering community.

NP-hardness and *NP*-completeness are originally defined to express the inherent difficulty of decision-type problems, not for approximation-type problems. Although approximation complexity is an important and active research area in the theory of computation (Papadimitriou 1995), most of the classical results are on decision-type problems. Many robust control-related problems can be formulated as "Check whether $\gamma < 1$," which is a decision-type problem. Approximate value of γ may not be always good enough to "solve" the problem, i.e., to decide about robust stability. For certain other engineering applications for which approximate values of optimization problems are good enough to "solve" the problem, the complexity of a decision problem may not be very relevant. For example, in a minimum effort control problem, usually there may be no point in computing the exact value of the minimum, because good approximations will

be just fine for most cases. However, for a robust control problem, a result like $\gamma = 0.99 \pm 0.02$ may be not enough to decide about robust stability, although the approximation error is about 2 % only. Basically, both the conservativeness of the current tractability definition and the differences between decision- and approximation-type problems should be always kept in mind when interpreting computational complexity results reported here as well as in the literature.

In this subsection, and in the next one, we will consider decision problems only. The class P corresponds to decision problems which can be solved by a Turing machine with a suitable program in polynomial time (Garey and Johnson 1979). This is interpreted as decision problems which have polynomial-time solution algorithms. The definition of the class NP is more technical and involves nondeterministic Turing machines (Garey and Johnson 1979). It may be interpreted as the class of decision problems for which the truth of the problem can be verified in polynomial time. It is currently unknown whether P and NP are equal or not. This is a major open problem, and the importance of it in the theory of computation is comparable to the importance of Riemann hypothesis in number theory.

A problem is NP-complete if it is NP and every NP problem polynomially reduces to it (Garey and Johnson 1979). For an NP-complete problem, being in P is equivalent to $P = NP$. There are literally hundreds of such problems, and it is generally argued that since after several years of research nobody was able to develop a polynomial-time algorithm for these NP-complete problems, there is probably no such algorithm, and most likely $P \neq NP$. Although current evidence is more toward $P \neq NP$, this does not constitute a formal proof, and the history of mathematics and science is full of surprising discoveries.

A problem (not necessarily NP) is called NP-hard if and only if there is an NP-complete problem which is polynomial time reducible to it (Garey and Johnson 1979). Being NP-hard is sometimes called being intractable and means that unless $P = NP$, which is considered to be very unlikely by a group of researchers, no

polynomial-time solution algorithm can be developed. All NP-complete problems are also NP-hard, but they are only as "hard" as any other problem in the set of NP-complete problems.

The first known NP-complete problem is SATISFIABILITY (Cook 1971). In this problem, there is a single Boolean equation with several variables, and we would like to test whether there is an assignment to these variables which make the Boolean expression true. This important discovery enabled proofs of NP-completeness or NP-hardness of several other problems by using simple polynomial reduction techniques only (Garey and Johnson 1979). Among these, quadratic programming is an important one and led to the discovery of several other complexity results in robust control theory. The quadratic programming (QP) can be defined as

$$q = \min_{Ax \le b} x^T Q x + c^T x,$$

more precisely testing whether $q < 1$ or not (decision version). When the matrix Q is positive definite, convex optimization techniques can be used; however, the general version of the problem is NP-hard.

A related problem is linear programming (LP)

$$q = \min_{Ax \le b} c^T x,$$

which is used in certain robust control problems (Dahleh and Diaz-Bobillo 1994) and has a quite interesting status. Simplex method (Dantzig 1963) is a very popular computational technique for LP and is known to have polynomial-time complexity on the "average" (Smale 1983). However, there are examples where the simplex method requires exponentially growing number of steps with the problem size (Klee and Minty 1972). In 1979, Khachiyan proposed the ellipsoid algorithm for LP, which was the first known polynomial-time approximation algorithm (Schrijver 1998). Because of the nature of the problem, one can answer the decision version of LP in polynomial time by using the ellipsoid algorithm for approximation and stopping when the error is below a certain level.

But all of these results are for standard Turing machines with input parameters restricted to rational numbers. An interesting open problem is whether LP admits a polynomial algorithm in the real number model of computation.

Complexity of Certain Robust Control Problems

There are several computational complexity results for robust control problems (see Blondel and Tsitsiklis (2000b) for a more detailed survey). Here we summarize some of the key results on interval matrices and structured singular values.

Kharitonov theorem is about robust Hurwitz stability of polynomials with coefficients restricted to intervals (Kharitonov 1978). The problem is known to have a surprisingly simple solution; however, the matrix version of the problem has a quite different nature. If we have a matrix family

$$\mathcal{A} = \big\{ A \in \mathbb{R}^{n \times n} : \alpha_{i,j} \le A_{i,j} \le \beta_{i,j}, \\ i, j = 1, \ldots, n \big\}, \quad (1)$$

where $\alpha_{i,j}, \beta_{i,j}$ are given constants for $i, j = 1, \ldots, n$, then it is called an interval matrix. Such matrices do occur in descriptions of uncertain dynamical systems. The following two stability problems about interval matrices are known to be *NP*-hard:

Interval Matrix Problem 1 (IMP1): Decide whether a given interval matrix, $\mathcal{A}$, is robust Hur- witz stable or not. Namely, check whether all members of $\mathcal{A}$ are Hurwitz-stable matrices, i.e., all eigenvalues are in open left half plane.

Interval Matrix Problem 2 (IMP2): Decide whether a given interval matrix, $\mathcal{A}$, has a Hurwitz-stable matrix or not. Namely, check whether there exists at least one matrix in $\mathcal{A}$ which is Hurwitz stable.

For a proof of *NP*-hardness of IMP1, see Poljak and Rohn (1993) and Nemirovskii (1993), and for a proof of IMP2, see Blondel and Tsitsiklis (1997).

Another important problem is related to structured singular values (SSV) and linear fractional transformations (LFT), which are mainly used to study systems which have component-level uncertainties (Packard and Doyle 1993). Basically, bounds on the component-level uncertainties are given, and we would like to check whether the system is robustly stable or not. This is known to be *NP*-hard.

Structured Singular Value Problem (SSVP): Given a matrix M and uncertainty structure $\mathbf{\Delta}$, check whether the structured singular value $\mu_{\mathbf{\Delta}}(M) < 1$.

This is proved to be *NP*-hard for real, and mixed, uncertainty structures (Braatz et al. 1994), as well as for complex uncertainties with no repetitions (Toker and Ozbay 1996, 1998).

Approximation Complexity

Decision version of QP is *NP*-hard, but approximation of QP is quite "difficult" as well. An approximation is called a μ-approximation if the absolute value of the error is bounded by μ times the absolute value of max–min of the function. The following is a classical result on QP (Bellare and Rogaway 1995): Unless $P = NP$, QP does not admit a polynomial-time μ-approximation algorithm even for $\mu < 1/3$. This is sometimes informally stated as "QP is *NP*-hard to approximate." Much work is needed toward similar results on robustness margin and related optimization problems of the classical robust control theory.

An interesting case is the complex structured singular value computation with no repeated uncertainties. There is a convex relaxation of the problem, the standard upper bound $\overline{\mu}$, which is known to result in quite tight approximations for most cases of the original problem (Packard and Doyle 1993). However, despite strong numerical evidence, a formal proof of "good approximation for most cases" result is not available. We also do not have much theoretical information about how hard it is to approximate the complex structured singular value. For example, it is not known whether it admits a polynomial-time approximation algorithm with error bounded by, say, 5 %. In summary, much work needs to be done in these

directions for many other robust control problems whose decision versions are *NP*-hard.

Summary and Future Directions

The study of the "Is $P \neq NP$?" question turned out to be a quite difficult one. Researchers agree that really new and innovative tools are needed to study this problem. On one other extreme, one can question whether we can really say something about this problem within the Zermelo-Fraenkel (ZF) set theory or will it have a status similar to axiom of choice (AC) and the continuum hypothesis (CH) where we can neither refute nor provide a proof (Aaronson 1995). Therefore, the question may be indeed much deeper than we thought, and standard axioms of today's mathematics may not be enough to provide an answer. As for any such problem, we can still hope that in the future, new "self-evident" axioms may be discovered, and with the help of them, we may provide an answer.

All of the complexity results mentioned here are with respect to the standard Turing machine which is a simplified model of today's digital computers. Depending on the progress in science, engineering, and technology, if superior computation machines can be constructed, then some of the abovementioned problems can be solved much faster on these devices, and current results/problems of the theory of computation may no longer be of great importance or relevance for engineering and scientific applications. In such a case, one may also need to revise definitions of the terms algorithm, tractable, etc., according to these new devices.

Currently, there are several *NP*-hardness results about robust control problems, mostly *NP*-hardness of decision problems about robustness. However, much work is needed on the approximation complexity and conservatism of various convex relaxations of these problems. Even if a robust stability problem is *NP*-hard, a polynomial-time algorithm to estimate robustness margin with, say, 5 % error is not ruled out with the *NP*-hardness of the decision version of the problem. Indeed, a polynomial-time

and 5 % error-bounded result will be of great importance for practicing engineers. Therefore, such directions should also be studied, and various meaningful alternatives, like being polynomial time on the average or for most of cases or anything which makes sense for a practicing engineer, should be considered as an alternative direction.

In summary, computational complexity theory guides research on the development of algorithms, indicating which directions are dead ends and which directions are worth to investigate.

Cross-References

▶ Optimization Based Robust Control
▶ Robust Control in Gap Metric
▶ Robust Fault Diagnosis and Control
▶ Robustness Issues in Quantum Control
▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Bibliography

Aaronson S (1995) Is P versus NP formally independent? Technical report 81, EATCS

Bellare M, Rogaway P (1995) The complexity of approximating a nonlinear program. Math Program 69:429–441

Blondel VD, Megretski A (2004) Unsolved problems in mathematical systems and control theory. Princeton University Press, Princeton

Blondel VD, Nesterov Y (2005) Computationally efficient approximations of the joint spectral radius. SIAM J Matrix Anal 27:256–272

Blondel VD, Tsitsiklis JN (1997) NP-hardness of some linear control design problems. SIAM J Control Optim 35:2118–2127

Blondel VD, Tsitsiklis JN (2000a) The boundedness of all products of a pair of matrices is undecidable. Syst Control Lett 41:135–140

Blondel VD, Tsitsiklis JN (2000b) A survey of computational complexity results in systems and control. Automatica 36:1249–1274

Blondel VD, Sontag ED, Vidyasagar M, Willems JC (1999) Open problems in mathematical systems and control theory. Springer, London

Blondel VD, Bournez O, Koiran P, Tsitsiklis JN (2001) The stability of saturated linear dynamical systems is undecidable. J Comput Syst Sci 62:442–462

Blondel VD, Theys J, Vladimirov AA (2003) An elementary counterexample to the finiteness conjecture. SIAM J Matrix Anal 24:963–970

Bousch T, Mairesse J (2002) Asymptotic height optimization for topical IFS, Tetris heaps and the finiteness conjecture. J Am Math Soc 15:77–111

Braatz R, Young P, Doyle J, Morari M (1994) Computational complexity of μ calculation. IEEE Trans Autom Control 39:1000–1002

Chen G, Church DA, Englert BG, Henkel C, Rohwedder B, Scully MO, Zubairy MS (2006) Quantum computing devices. Chapman and Hall/CRC, Boca Raton

Cook S (1971) The complexity of theorem proving procedures. In: Proceedings of the third annual ACM symposium on theory of computing, Shaker Heights, pp 151–158

Dahleh MA, Diaz-Bobillo I (1994) Control of uncertain systems. Prentice Hall, Englewood Cliffs

Dantzig G (1963) Linear programming and extensions. Princeton University Press, Princeton

Davis M (1985) Computability and unsolvability. Dover

Garey MR, Johnson DS (1979) Computers and intractability, a guide to the theory of NP-completeness. W. H. Freeman, San Francisco

Hopcroft JE, Motwani R, Ullman JD (2001) Introduction to automata theory, languages, and computation. Addison Wesley, Boston

Kaye P, Laflamme R, Mosca M (2007) An introduction to quantum computing. Oxford University Press, Oxford

Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. Differentsial'nye Uravneniya 14: 2086–2088

Klee V, Minty GJ (1972) How good is the simplex algorithm? In: Inequalities III (proceedings of the third symposium on inequalities), Los Angeles. Academic, New York/London, pp 159–175

Knuth DE (1997) Art of computer programming, volume 2: seminumerical algorithms, 3rd edn. Addison-Wesley, Reading

Lagarias JC, Wang Y (1995) The finiteness conjecture for the generalized spectral radius of a set of matrices. Linear Algebra Appl 214:17–42

Lewis HR, Papadimitriou CH (1998) Elements of the theory of computation. Prentice Hall, Upper Saddle River

Matiyasevich YV (1993) Quantum computing devices. MIT

Nemirovskii A (1993) Several NP-hard problems arising in robust stability analysis. Math Control Signals Syst 6:99–105

Packard A, Doyle J (1993) The complex structured singular value. Automatica 29:71–109

Papadimitriou CH (1995) Computational complexity. Addison-Wesley/Longman, Reading

Poljak S, Rohn J (1993) Checking robust nonsingularity is NP-hard. Math Control Signals Syst 6:1–9

Rota GC, Strang G (1960) A note on the joint spectral radius. Proc Neth Acad 22:379–381

Schrijver A (1998) Theory of linear and integer programming. Wiley, Chichester

Sipser M (2006) Introduction to the theory of computation. Thomson Course Technology, Boston

Smale S (1983) On the average number of steps in the simplex method of linear programming. Math Program 27:241–262

Toker O, Ozbay H (1996) Complexity issues in robust stability of linear delay differential systems. Math Control Signals Syst 9:386–400

Toker O, Ozbay H (1998) On the NP-hardness of the purely complex mu computation, analysis/synthesis, and some related problems in multidimensional systems. IEEE Trans Autom Control 43:409–414

Turing AM (1936) On computable numbers, with an application to the Entscheidungsproblem. Proc Lond Math Soc 42:230–265

Computer-Aided Control Systems Design: Introduction and Historical Overview

Andreas Varga
Institute of System Dynamics and Control, German Aerospace Center, DLR Oberpfaffenhofen, Wessling, Germany

Synonyms

CACSD

Abstract

Computer-aided control system design (CACSD) encompasses a broad range of Methods and tools and technologies for system modelling, control system synthesis and tuning, dynamic system analysis and simulation, as well as validation and verification. The domain of CACSD enlarged progressively over decades from simple collections of algorithms and programs for control system analysis and synthesis to comprehensive tool sets and user-friendly environments supporting all aspects of developing and deploying advanced control systems in various application fields. This entry gives a brief introduction to CACSD and reviews

the evolution of key concepts and technologies underlying the CACSD domain. Several cornerstone achievements in developing reliable numerical algorithms; implementing robust numerical software; and developing sophisticated integrated modelling, simulation, and design environments are highlighted.

Keywords

CACSD; Modelling; Numerical analysis; Simulation; Software tools

Introduction

To design a control system for a plant, a typical *computer-aided control system design* (CACSD) work flow comprises several interlaced activities.

Model building is often a necessary first step consisting in developing suitable mathematical models to accurately describe the plant dynamical behavior. High-fidelity physical plant models obtained, for example, by using the first principles of modelling, primarily serve for analysis and validation purposes using appropriate simulation techniques. These dynamic models are usually defined by a set of *ordinary differential equations* (ODEs), *differential algebraic equation* (DAEs), or *partial differential equations* (PDEs). However, for control system synthesis purposes simpler models are required, which are derived by simplifying high-fidelity models (e.g., by linearization, discretization, or model reduction) or directly determined in a specific form from input-output measurement data using system identification techniques. Frequently used synthesis models are continuous or discrete-time *linear time-invariant* (LTI) models describing the nominal behavior of the plant in a specific operating point. The more accurate *linear parameter varying* (LPV) models may serve to account for uncertainties due to various performed approximations, nonlinearities, or varying model parameters.

Simulation of dynamical systems is a closely related activity to modelling and is concerned with performing virtual experiments on a given plant model to analyze and predict the dynamic behavior of a physical plant. Often, modelling and simulation are closely connected parts of dedicated environments, where specific classes of models can be built and appropriate simulation methods can be employed. Simulation is also a powerful tool for the validation of mathematical models and their approximations. In the context of CACSD, simulation is frequently used as a control system tuning-aid, as, for example, in an optimization-based tuning approach using time-domain performance criteria.

System analysis and synthesis are concerned with the investigation of properties of the underlying synthesis model and in the determination of a control system which fulfills basic requirements for the closed-loop controlled plant, such as stability or various time or frequency response requirements. The analysis also serves to check existence conditions for the solvability of synthesis problems, according to established design methodologies. An important synthesis goal is the guarantee of the performance robustness. To achieve this goal, robust control synthesis methodologies often employ optimization-based parameter tuning in conjunction with worst-case analysis techniques. A rich collection of reliable numerical algorithms are available to perform such analysis and synthesis tasks. These algorithms form the core of CACSD and their development represented one of the main motivations for CACSD-related research.

Performance robustness assessment of the resulting closed-loop control system is a key aspect of the verification and validation activity. For a reliable assessment, simulation-based worst-case analysis represents, often, the only way to prove the performance robustness of the synthesized control system in the presence of parametric uncertainties and variabilities.

Development of CACSD Tools

The development of CACSD tools for system analysis and synthesis started around 1960, immediately after general-purpose digital

computers, and new programming languages became available for research and development purposes. In what follows, we give a historical survey of these developments in the main CACSD areas.

Modelling and Simulation Tools

Among the first developments were modelling and simulation tools for continuous-time systems described by differential equations based on dedicated simulation languages. Over 40 continuous-system simulation languages had been developed as of 1974 (Nilsen and Karplus 1974), which evolved out of attempts at digitally emulating the behavior of widely used analog computers before 1960. A notable development in this period was the CSSL standard (Augustin et al. 1967), which defined a system as an interconnection of blocks corresponding to operators which emulated the main analog simulation blocks and implied the integration of the underlying ODEs using suitable numerical methods. For many years, the ACSL preprocessor to Fortran (Mitchel and Gauthier 1976) was one of the most successful implementations of the CSSL standard.

A turning point was the development of graphical user interfaces allowing graphical block diagram-based modelling. The most important developments were SystemBuild (Shah et al. 1985) and SIMULAB (later marketed as Simulink) (Grace 1991). Both products used a customizable set of block libraries and were seamlessly integrated in, respectively, MATRIXx and MATLAB, two powerful interactive matrix computation environments (see below). SystemBuild provided several advanced features such as event management, code generation, and DAE-based modelling and simulation. Simulink excelled from the beginning with its intuitive, easy-to-use user interface. Recent extensions of Simulink allow the modelling and simulation of hybrid systems, code generation for real-time applications, and various enhancements of the model building process (e.g., object-oriented modelling).

The object-oriented paradigm for system modelling was introduced with Dymola (Elmqvist 1978) to support physical system modelling based on interconnections of subsystems. The underlying modelling language served as the basis of the first version of Modelica (Elmquist et al. 1997), a modern equation-based modelling language which was the result of a coordinated effort for the unification and standardization of expertise gained over many years with object-oriented physical modelling. The latest developments in this area are comprehensive model libraries for different application domains such as mechanical, electrical, electronic, hydraulic, thermal, control, and electric power systems. Notable commercial front-ends for Modelica are Dymola, MapleSim, and SystemModeler, where the last two are tightly integrated in the symbolic computation environments Maple and Mathematica, respectively.

Numerical Software Tools

The computational tools for CACSD rely on many numerical algorithms whose development and implementation in computer codes was the primary motivation of this research area since its beginnings. The Automatic Synthesis Program (ASP) developed in 1966 (Kalman and Englar 1966) was implemented in FAP (Fortran Assembly Program) and ran only on an IBM 7090–7094 machine. The Fortran II version of ASP (known as FASP) can be considered to be the first collection of computational CACSD tools ported to several mainframe computers. Interestingly, the linear algebra computations were covered by only three routines (diagonal decomposition, inversion, and pseudoinverse), and no routines were used for eigenvalue or polynomial roots computation. The main analysis and synthesis functions covered the sampled-data discretization (via matrix exponential), minimal realization, time-varying Riccati equation solution for quadratic control, filtering, and stability analysis. The FASP itself performed the required computational sequences by interpreting simple commands with parameters. The extensive documentation containing a detailed description of algorithmic approaches and many examples marked the starting point of an intensive research on algorithms and numerical software, which culminated in the development of the

high-performance control and systems library SLICOT (Benner et al. 1999; Huffel et al. 2004). In what follows, we highlight the main achievements along this development process.

The direct successor of FASP is the Variable Dimension Automatic Synthesis Program (VASP) (implemented in Fortran IV on IBM 360) (White and Lee 1971), while a further development was ORACLS (Armstrong 1978), which included several routines from the newly developed eigenvalue package EISPACK (Garbow et al. 1977; Smith et al. 1976) as well as solvers for linear (Lyapunov, Sylvester) and quadratic (Riccati) matrix equations. From this point, the mainstream development of numerical algorithms for linear system analysis and synthesis closely followed the development of algorithms and software for numerical linear algebra. A common feature of all subsequent developments was the extensive use of robust linear algebra software with the Basic Linear Algebra Subprograms (BLAS) (Lawson et al. 1979) and the Linear Algebra Package (LINPACK) for solving linear systems (Dongarra et al. 1979). Several control libraries have been developed almost simultaneously, relying on the robust numerical linear algebra core software formed of BLAS, LINPACK, and EISPACK. Notable examples are RASP (based partly on VASP and ORACLS) (Grübel 1983) – developed originally by the University of Bochum and later by the German Aerospace Center (DLR); BIMAS (Varga and Sima 1985) and BIMASC (Varga and Davidoviciu 1986) – two Romanian initiatives; and SLICOT (Boom et al. 1991) – a Benelux initiative of several universities jointly with the Numerical Algorithm Group (NAG).

The last development phase was marked by the availability of the Linear Algebra Package (LAPACK) (Anderson et al. 1992), whose declared goal was to make the widely used EISPACK and LINPACK libraries run efficiently on shared memory vector and parallel processors. To minimize the development efforts, several active research teams from Europe started, in the framework of the NICONET project, a concentrated effort to develop a high-performance numerical software library

for CACSD as a new significantly extended version of the original SLICOT. The goals of the new library were to cover the main computational needs of CACSD, by relying exclusively on LAPACK and BLAS, and to guarantee similar numerical performance as that of the LAPACK routines. The software development used rigorous standards for implementation in Fortran 77, modularization, testing, and documentation (similar to that used in LAPACK). The development of the latest versions of RASP and SLICOT eliminated practically any duplication of efforts and led to a library which contained the best software from RASP, SLICOT, BIMAS, and BIMASC. The current version of SLICOT is fully maintained by the NICONET association (http://www.niconet-ev.info/en/) and serves as basis for implementing advanced computational functions for CACSD in interactive environments as MATLAB (http://www.mathworks.com), Maple (http://www.maplesoft.com/products/maple/), Scilab (http://www.scilab.org/) and Octave (http://www.gnu.org/software/octave/).

Interactive Tools

Early experiments during 1970–1985 included the development of several interactive CACSD tools employing menu-driven interaction, question-answer dialogues, or command languages. The April 1982 special issue of IEEE Control Systems Magazine was dedicated to CACSD environments and presented software summaries of 20 interactive CACSD packages. This development period was marked by the establishment of new standards for programming languages (Fortran 77, C), availability of high-quality numerical software libraries (BLAS, EISPACK, LINPACK, ODEPACK), transition from mainframe computers to minicomputers, and finally to the nowadays-ubiquitous personal computers as computing platforms, spectacular developments in graphical display technologies, and application of sound programming paradigms (e.g., strong data typing).

A remarkable event in this period was the development of MATLAB, a command language-based interactive matrix laboratory (Moler 1980).

The original version of MATLAB was written in Fortran 77. It was primarily intended as a student teaching tool and provided interactive access to selected subroutines from LINPACK and EISPACK. The tool circulated for a while in the public domain, and its high flexibility was soon recognized. Several CACSD-oriented commercial clones have been implemented in the C language, the most important among them being MATRIXx (Walker et al. 1982) and PC-MATLAB (Moler et al. 1985).

The period after 1985 until around 2000 can be seen as a consolidation and expansion period for many commercial and noncommercial tools. In an inventory of CACSD-related software issued by the Benelux Working Group on Software (WGS) under the auspices of the IEEE Control Systems Society, there were in 1992 in active development 70 stand-alone CACSD packages, 21 tools based on or similar to MATLAB, and 27 modelling/simulation environments. It is interesting to look more closely at the evolutions of the two main players MATRIXx and MATLAB, which took place under harshly competitive conditions.

MATRIXx with its main components Xmath, SystemBuild, and AutoCode had over many years a leading role (especially among industrial customers), excelling with a rich functionality in domains such as system identification, control system synthesis, model reduction, modelling, simulation, and code generation. After 2003, MATRIXx (http://www.ni.com/matrixx/) became a product of the National Instruments Corporation and complements its main product family LabView, a visual programming language-based system design platform and development environment (http://www.ni.com/labview).

MATLAB gained broad academic acceptance by integrating many new methodological developments in the control field into several control-related toolboxes. MATLAB also evolved as a powerful programming language, which allows easy object-oriented manipulation of different system descriptions via operator overloading. At present, the CACSD tools of MATLAB and Simulink represent the industrial and academic standard for CACSD. The existing CACSD tools are constantly extended and enriched with new model classes, new computational algorithms (e.g., structure-exploiting eigenvalue computations based on SLICOT), dedicated graphical user interfaces (e.g., tuning of PID controllers or control-related visualizations), advanced robust control system synthesis, etc. Also, many third-party toolboxes contribute to the wide usage of this tool.

Basic CACSD functionality incorporating symbolic processing techniques and higher precision computations is available in the Maple product MapleSim Control Design Toolbox as well as in the Mathematica Control Systems product. Free alternatives to MATLAB are the MATLAB-like environments Scilab, a French initiative pioneered by INRIA, and Octave, which has recently added some CACSD functionality.

Summary and Future Directions

The development and maintenance of integrated CACSD environments, which provide support for all aspects of the CACSD cycle such as modelling, design, and simulation, involve sustained, strongly interdisciplinary efforts. Therefore, the CACSD tool development activities must rely on the expertise of many professionals covering such diverse fields as control system engineering, programming languages and techniques, man-machine interaction, numerical methods in linear algebra and control, optimization, computer visualization, and model building techniques. This may explain why currently only a few of the commercial developments of prior years are still in use and actively maintained/developed. Unfortunately, the number of actively developed noncommercial alternative products is even lower. The dominance of MATLAB, as a de facto standard for both industrial and academic usage of integrated tools covering all aspects of the broader area of *computer-aided control engineering* (CACE), cannot be overseen.

The new trends in CACSD are partly related to handling more complex applications, involving time-varying (e.g., periodic, multi-rate sampled-data, and differential algebraic) linear dynamic systems, nonlinear systems with many parametric uncertainties, and large-scale models (e.g., originating from the discretization of PDEs). To address many computational aspects of model building (e.g., model reduction of large order systems), optimization-based robust controller tuning using multiple-model approaches, or optimization-based robustness assessment using global-optimization techniques, parallel computation techniques allow substantial savings in computational times and facilitate addressing computational problems for large-scale systems. A topic which needs further research is the exploitation of the benefits of combining numerical and symbolic computations (e.g., in model building and manipulation).

Cross-References

- ▶ Interactive Environments and Software Tools for CACSD
- ▶ Model Building for Control System Synthesis
- ▶ Model Order Reduction: Techniques and Tools
- ▶ Multi-domain Modeling and Simulation
- ▶ Optimization-Based Control Design Techniques and Tools
- ▶ Robust Synthesis and Robustness Analysis Techniques and Tools
- ▶ Validation and Verification Techniques and Tools

Recommended Reading

The historical development of CACSD concepts and techniques was the subject of several articles in reference works Rimvall and Jobling (1995) and Schmid (2002). A selection of papers on numerical algorithms underlying the development of CACSD appeared in Patel et al. (1994). The special issue No. 2/2004 of the IEEE Control Systems Magazine on *Numerical Awareness in Control* presents several surveys on different

aspects of developing numerical algorithms and software for CACSD.

The main trends over the last three decades in CACSD-related research can be followed in the programs/proceedings of the biannual IEEE Symposia on CACSD from 1981 to 2013 (partly available at http://ieeexplore.ieee.org) as well as of the triennial IFAC Symposia on CACSD from 1979 to 2000. Additional information can be found in several CACSD-focused survey articles and special issues (e.g., No. 4/1982; No. 2/2000) of the IEEE Control Systems Magazine.

Bibliography

Anderson E, Bai Z, Bishop J, Demmel J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S, Sorensen D (1992) LAPACK user's guide. SIAM, Philadelphia

Armstrong ES (1978) ORACLS – a system for linear-quadratic Gaussian control law design. Technical paper 1106 96-1, NASA

Augustin DC, Strauss JC, Fineberg MS, Johnson BB, Linebarger RN, Sansom FJ (1967) The SCi continuous system simulation language (CSSL). Simulation 9:281–303

Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – a subroutine library in systems and control theory. In: Datta BN (ed) Applied and computational control, signals and circuits, vol 1. Birkhäuser, Boston, pp 499–539

Dongarra JJ, Moler CB, Bunch JR, Stewart GW (1979) LINPACK user's guide. SIAM, Philadelphia

Elmquist H et al (1997) Modelica – a unified object-oriented language for physical systems modeling (version 1). http://www.modelica.org/documents/Modelica1.pdf

Elmqvist H (1978) A structured model language for large continuous systems. PhD thesis, Department of Automatic Control, Lund University, Sweden

Garbow BS, Boyle JM, Dongarra JJ, Moler CB (1977) Matrix eigensystem routines – EISPACK guide extension. Springer, Heidelberg

Grace ACW (1991) SIMULAB, an integrated environment for simulation and control. In: Proceedings of American Control Conference, Boston, pp 1015–1020

Grübel G (1983) Die regelungstechnische Programmbibliothek RASP. Regelungstechnik 31:75–81

Kalman RE, Englar TS (1966) A user's manual for the automatic synthesis program (program C). Technical report CR-475, NASA

Lawson CL, Hanson RJ, Kincaid DR, Krogh FT (1979) Basic linear algebra subprograms for Fortran usage. ACM Trans Math Softw 5:308–323

Mitchel EEL, Gauthier JS (1976) Advanced continuous simulation language (ACSL). Simulation 26: 72–78

Moler CB (1980) MATLAB users' guide. Technical report, Department of Computer Science, University of New Mexico, Albuquerque

Moler CB, Little J, Bangert S, Kleinman S (1985) PC-MATLAB, users' guide, version 2.0. Technical report, The MathWorks Inc., Sherborn

Nilsen RN, Karplus WJ (1974) Continuous-system simulation languages: a state-of-the-art survey. Math Comput Simul 16:17–25. doi:http://dx.doi.org/10.1016/S0378-4754(74)80003-0

Patel RV, Laub AJ, Van Dooren P (eds) (1994) Numerical linear algebra techniques for systems and control. IEEE, Piscataway

Rimvall C, Jobling CP (1995) Computer-aided control systems design. In: Levine WS (ed) The control handbook. CRC, Boca Raton, pp 429–442

Schmid C (2002) Computer-aided control system engineering tools. In: Unbehauen H (ed) Control systems, robotics and automation. http://www.eolss.net/outlinecomponents/Control-Systems-Robotics-Automation.aspx

Shah CS, Floyd MA, Lehman LL (1985) MATRIXx: control design and model building CAE capabilities. In: Jamshidi M, Herget CJ (eds) Advances in computer aided control systems engineering. North-Holland/Elsevier, Amsterdam, pp 181–207

Smith BT, Boyle JM, Dongarra JJ, Garbow BS, Ikebe Y, Klema VC, Moler CB (1976) Matrix eigensystem routines – EISPACK guide. Lecture notes in computer science, vol 6, 2nd edn. Springer, Berlin/New York

van den Boom A, Brown A, Geurts A, Hammarling S, Kool R, Vanbegin M, Van Dooren P, Van Huffel S (1991) SLICOT, a subroutine library in control and systems theory. In: Preprints of 5th IFAC/IMACS symposium of CADCS'91, Swansea. Pergamon Press, Oxford, pp 89–94

Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. Control Syst Mag 24:60–76

Varga A, Davidoviciu A (1986) BIMASC – a package of Fortran subprograms for analysis, modelling, design and simulation of control systems. In: Hansen NE, Larsen PM (eds) Preprints of 3rd IFAC/IFIP International Symposium on Computer Aided Design in Control and Engineering (CADCE'85), Copenhagen. Pergamon Press, Oxford, pp 151–156

Varga A, Sima V (1985) BIMAS – a basic mathematical package for computer aided systems analysis and design. In: Gertler J, Keviczky L (eds) Preprints of 9th IFAC World Congress, Hungary, vol 8, pp 202–207

Walker R, Gregory C, Shah S (1982) MATRIXx: a data analysis, system identification, control design and simulation package. Control Syst Mag 2:30–37

White JS, Lee HQ (1971) Users manual for the variable automatic synthesis program (VASP). Technical memorandum TM X-2417, NASA

Consensus of Complex Multi-agent Systems

Fabio Fagnani
Dipartimento di Scienze Matematiche 'G.L. Lagrange', Politecnico di Torino, Torino, Italy

Abstract

This entry provides a broad overview of the basic elements of consensus dynamics. It describes the classical Perron-Frobenius theorem that provides the main theoretical tool to study the convergence properties of such systems. Classes of consensus models that are treated include simple random walks on grid-like graphs and in graphs with a bottleneck, consensus on graphs with intermittently randomly appearing edges between nodes (gossip models), and models with nodes that do not modify their state over time (stubborn agent models). Application to cooperative control, sensor networks, and socioeconomic models are mentioned.

Keywords

Consensus; Electrical networks; Gossip model; Spectral gap; Stubborn agents

Multi-agent Systems and Consensus

Multi-agent systems constitute one of the fundamental paradigms of science and technology of the present century (Castellano et al. 2009; Strogatz 2003). The main idea is that of creating complex dynamical evolutions from the interactions of many simple units. Indeed such collective behaviors are quite evident in biological and social systems and were indeed considered in earlier times. More recently, the digital revolution and the miniaturization in electronics have made possible the creation of man-made complex architectures of interconnected simple devices (computers, sensors, cameras). Moreover, the creation of the Internet has opened a totally

new form of social and economic aggregation. This has strongly pushed towards a systematic and deep study of multi-agent dynamical systems. Mathematically they typically consist of a graph where each node possesses a state variable; states are coupled at the dynamical level through dependences determined by the edges in the graph. One of the challenging problems in the field of multi-agent systems is to analyze the emergence of complex collective phenomena from the interactions of the units which are typically quite simple. Complexity is typically the outcome of the topology and the nature of interconnections.

Consensus dynamics (also known as average dynamics) (Carli et al. 2008; Jadbabaie et al. 2003) is one of the most popular and simplest multi-agent dynamics. One convenient way to introduce it is with the language of social sciences. Imagine that a number of independent units possess an information represented by a real number, for instance, such number can represent an opinion on a given fact. Units interact and change their opinion by averaging with the opinions of other units. Under certain assumptions, this will lead the all community to converge to a consensus opinion which takes into consideration all the initial opinion of the agents. In social sciences, empiric evidences (Galton 1907) have shown how such aggregate opinion may give a very good estimation of unknown quantities: such phenomenon has been proposed in the literature as wisdom of crowds (Surowiecki 2004).

Consensus Dynamics, Graphs, and Stochastic Matrices

Mathematically, consensus dynamics are special linear dynamical systems of type

$$x(t+1) = Px(t) \qquad (1)$$

where $x(t) \in \mathbb{R}^\mathcal{V}$ and $P \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ is a *stochastic matrix* (e.g., a matrix with nonnegative elements such that every row sums to 1). $\mathcal{V}$ represents the finite set of units (agents) in the network and $x(t)_v$ is to be interpreted has the state (opinion) of agent v at time t. Equation (1) implies that states of agents at time $t+1$ are convex

combinations of the components of $x(t)$: this motivates the term averaging dynamics. Stochastic matrices owe their name to their use in probability: the term P_{vw} can be interpreted as the probability of making a jump in the graph from state v to state w. In this way you construct what is called a random walk on the graph $\mathcal{G}$.

The network structure is hidden in the nonzero pattern of P. Indeed we can associate to P a graph: $\mathcal{G}_P = (\mathcal{V}, \mathcal{E}_P)$ where the set of edges is given by $\mathcal{E}_P := \{(u, v) \in \mathcal{V} \times \mathcal{V} \mid P_{uv} > 0\}$. Elements in $\mathcal{E}_P$ represent the communication edges among the units: if $(u, v) \in \mathcal{E}_P$, it means that unit u has access to the state of unit v. Denote by $\mathbb{1} \in \mathbb{R}^\mathcal{V}$ the all 1's vector. Notice that $P\mathbb{1} = \mathbb{1}$: this shows that once the states of units are at consensus, they will no longer evolve. Will the dynamics always converge to a consensus point?

Remarkably, some of the key properties of P responsible for the transient and asymptotic behavior of the linear system (1) are determined by the connectivity properties of the underlying graph $\mathcal{G}_P$. We recall that, given two vertices $u, v \in \mathcal{V}$, a *path* (of length l) from u to v in $\mathcal{G}_P$ is any sequence of vertices $u = u_1, u_2, \ldots, u_{l+1} = v$ such that $(u_i, u_{i+1}) \in \mathcal{E}_P$ for every $i = 1, \ldots, s$. $\mathcal{G}_P$ is said to be *strongly connected* if for any pair of vertices $u \neq v$ in $\mathcal{V}$ there is a path in $\mathcal{G}_P$ connecting u to v. The *period* of a node u is defined as the greatest common divisor of the lengths of all closed paths from u to u. In the strongly connected graph, all nodes have the same period, and the graph is called aperiodic if such a period is 1. If x is a vector, we will use the notation x^* to denote its transpose. If A is a finite set, $|A|$ denotes the number of elements in A. The following classical result holds true (Gantmacher 1959):

Theorem 1 (Perron-Frobenius) *Assume that $P \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ is such that $\mathcal{G}_P$ is strongly connected and aperiodic. Then,*

1. *1 is an algebraically simple eigenvalue of P.*
2. *There exists a (unique) probability vector $\pi \in \mathbb{R}^\mathcal{V}$ ($\pi_v > 0$ for all v and $\sum_v \pi_v = 1$) which is a left eigenvector for P, namely, $\pi^* P = \pi^*$.*
3. *All the remaining eigenvalues of P are of modulus strictly less than 1.*

A straightforward linear algebra consequence of this result is that $P^t \to \mathbb{1}\pi^*$ for $t \to +\infty$. This yields

$$\lim_{t \to +\infty} x(t) = \lim_{t \to +\infty} P^t x(0) = \mathbb{1}(\pi^* x(0)) \tag{2}$$

All agents' state are thus converging to the common value $\pi^* x(0)$, called *consensus point* which is a convex combination of the initial states with weights given by the invariant probability components.

If π is the uniform vector (i.e., $\pi_v = |\mathcal{V}|^{-1}$ for all units v), the common asymptotic value is simply the arithmetic mean of the initial states: all agents equally contribute to the final common state. A special case when this happens is when P is symmetric. The distributed computation of the arithmetic mean is an important step to solve estimation problems for sensor networks. As a specific example, consider the situation where there are N sensors deployed in a certain area and each of them makes a noisy measurement of a physical quantity x. Let $y_v = x + \omega_v$ be the measure obtained by sensor v, where ω_v is a zero mean Gaussian noise. It is well known that if noises are independent and identically distributed, the optimal mean square estimator of the quantity x given the entire set of measurements $\{y_v\}$ is exactly given by $\hat{x} = N^{-1} \sum_v y_v$. Other fields of application is in the control of cooperative autonomous vehicles (Fax and Murray 2004; Jadbabaie et al. 2003).

Basic linear algebra allows to study the rate of convergence to consensus. Indeed, if $\mathcal{G}_P$ is strongly connected and aperiodic, the matrix P has all its eigenvalues in the unit ball: 1 is the only eigenvalue with modulo equal to 1, while all the others have modulo strictly less than one. If we denote by $\rho_2 < 1$ the largest modulo of such eigenvalues (different from 1), we can show that $x(t) - \mathbb{1}(\pi^* x(0))$ converges exponentially to 0 as ρ_2^t. In the following, we will briefly refer to ρ_2 as to the *second eigenvalue* of P.

Examples and Large-Scale Analysis

In this section, we present some classical examples. Consider a strongly connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The *adjacency matrix* of $\mathcal{G}$ is a square matrix $A_\mathcal{G} \in \{0,1\}^{\mathcal{V} \times \mathcal{V}}$ such that $(A_\mathcal{G})_{uv} = 1$ iff $(u,v) \in \mathcal{E}$. $\mathcal{G}$ is said to be symmetric if $A_\mathcal{G}$ is symmetric. Given a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we can consider the stochastic matrix P given by $P_{uv} = d_u^{-1}(A_\mathcal{G})_{uv}$ where $d_u = \sum_v (A_\mathcal{G})_{uv}$ is the *degree* of node u. P is called the *simple random walk (SRW)* on $\mathcal{G}$: each agent gives the same weight to the state of its neighbors. Clearly, $\mathcal{G}_P = \mathcal{G}$. A simple check shows that $\pi_v = d_v/|\mathcal{E}|$ is the invariant probability for P. The consensus point is given by

$$\pi^* x(0) = |\mathcal{E}|^{-1} \sum_v d_v x(0)_v$$

Each node contributes with its initial state to this consensus with a weight which is proportional to the degree of the node. Notice that the SRW P is symmetric iff the graph is regular, namely, all units have the same degree.

We now present a number of classical examples based on families of graphs with larger and larger number of nodes N. In this setting, particularly relevant is to understand the behavior of the second eigenvalue ρ_2 as a function of N. Typically, one considers $\epsilon > 0$ fixed and solves the equation $\rho_2^t = \epsilon$. The solution $\tau = (\ln \rho_2^{-1})^{-1} \ln \epsilon^{-1}$ will be called the *convergence time*: it essentially represents the time needed to shrink of a factor ϵ the distance to consensus. Dependence of ρ_2 on N will also yield that τ will be a function of N. In the sequel, we will investigate such dependence for SRW's on certain classical families of graphs.

Example 1 (SRW on a complete graph) Consider the complete graph on the set $\mathcal{V}$: $K_\mathcal{V} := (\mathcal{V}, \mathcal{V} \times V)$ (also self loops are present). The SRW on $K_\mathcal{V}$ is simply given by $P = N^{-1}\mathbb{1}\mathbb{1}^*$ where $N = |\mathcal{V}|$. Clearly, $\pi = N^{-1}\mathbb{1}$. Eigenvalues of P are 1 with multiplicity 1 and 0 with multiplicity $N - 1$. Therefore, $\rho_2 = 0$. Consensus in this case

is achieved in just one step: $x(t) = N^{-1} \mathbb{1} \mathbb{1}^* x(0)$ for all $t \geq 1$.

Example 2 (SRW on a cycle graph) Consider the symmetric cycle graph $C_N = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{0, \ldots, N-1\}$ and $\mathcal{E} = \{(v, v+1), (v+1, v)\}$ where sum is mod N. The graph C_N is clearly strongly connected and is also aperiodic if N is odd. The corresponding SRW P has eigenvalues

$$\lambda_k = \cos \frac{2\pi k}{N}$$

Therefore, if N is odd, the second eigenvalue is given by

$$\rho_2 = \cos \frac{2\pi}{N} = 1 - 2\pi^2 \frac{1}{N^2} + o(N^{-2}) \qquad (3)$$
$$\text{for } N \to +\infty$$

while the corresponding convergence time is given by

$$\tau = (\ln \rho_2^{-1})^{-1} \ln \epsilon^{-1} \asymp N^2 \quad \text{for } N \to +\infty$$

Example 3 (SRW on toroidal grids) The toroidal d-grids C_n^d is formally obtained as a product of cycle graphs. The number of nodes is $N = n^d$. It can be shown that the convergence time behaves as

$$\tau \asymp N^{2/d} \quad \text{for } N \to +\infty$$

Convergence time exhibits a slower growth in N as the dimension d of the grid increases: this is intuitive since the increase in d determines a better connectivity of the graph and a consequently faster diffusion of information.

For a general stochastic matrix (even for SRW on general graphs), the computation of the second eigenvalue is not possible in closed form and can actually be also difficult from a numerical point of view. It is therefore important to develop tools for efficient estimation. One of these is based on the concept of bottleneck: if a graph can be splitted into two loosely connected parts, then consensus dynamics will necessarily exhibit a slow convergence.

Formally, given a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a subset of nodes $S \subseteq \mathcal{V}$, define e_S as the number of edges with at least one node in S and e_{SS} as the number of edges with both nodes in S. The bottleneck of S in G is defined as $\Phi(S) = e_{SS}/e_S$. Finally, the *bottleneck ratio* of $\mathcal{G}$ is defined as

$$\Phi_* := \min_{S : e_S/e \leq 1/2} \Phi(S)$$

where $e = |\mathcal{E}|$ is the number of edges in the graph.

Let P be the SRW on $\mathcal{G}$ and let ρ_2 be its second eigenvalue. Then,

Proposition 1 (Cheeger bound Levin et al. 2008)
$$1 - \rho_2 \leq 2\Phi_*. \qquad (4)$$

Example 4 (Graphs with a bottleneck) Consider two complete graphs with n nodes connected by just one edge. If S is the set of nodes of one of the two complete graphs, we obtain

$$\Phi(S) = \frac{1}{n^2 + 1}$$

Bound (4) implies that the convergence time is at least of the order of n^2 in spite of the fact that in each complete graph convergence would be in finite time!

Other Models

The systems studied so far are based on the assumptions that units all behave the same, and they share a common clock and update their state in a synchronous fashion. In this section, we discuss more general models.

Random Consensus Models
Regarding the assumption of synchronicity, it turns out to be unfeasible in many contexts. For instance, in the opinion dynamics modeling, it

is not realistic to assume that all interactions happen at the same time: agents are embedded in a physical continuous time, and interactions can be imagined to take place at random, for instance, in a pairwise fashion.

One of the most famous random consensus model is the gossip model. Fix a real number $q \in (0, 1)$ and a symmetric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. At every time instant t, an edge $(u, v) \in \mathcal{E}$ is activated with uniform probability $|\mathcal{E}|^{-1}$, and nodes u and v exchange their states and produce a new state according to the equations

$$x_u(t + 1) = (1 - q)x_u(t) + qx_v(t)$$
$$x_v(t + 1) = qx_u(t) + (1 - q)x_v(t)$$

The states of the other units remain unchanged.

Will this dynamics lead to a consensus? If the same edge is activated at every time instant, clearly consensus will not be achieved. However, it can be shown that, with probability one, consensus will be reached (Boyd et al. 2006).

Consensus Dynamics with Stubborn Agents

In this entry, we investigate consensus dynamics models where some of the agents do not modify their own state (stubborn agents). These systems are of interest in socioeconomic models (Acemoglu et al. 2013).

Consider a symmetric connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We assume a splitting $\mathcal{V} = \mathcal{S} \cup \mathcal{R}$ with the understanding that agents in S are *stubborn* agents not changing their state, while those in $\mathcal{R}$ are *regular* agents whose state modifies in time according to a SRW consensus dynamics, namely,

$$x_u(t + 1) = \frac{1}{d_u} \sum_{v \in \mathcal{V}} (A_{\mathcal{G}})_{uv} x_v(t), \quad \forall u \in \mathcal{R}$$

By assembling the state of the regular and of the stubborn agents in vectors denoted, respectively, as $x^{\mathcal{R}}(t)$ and $x^{\mathcal{S}}(t)$, dynamics can be recast in matrix form as

$$\begin{aligned} x^{\mathcal{R}}(t + 1) &= Q^{11}x^{\mathcal{R}}(t) + Q^{12}x^{\mathcal{S}}(t) \\ x^{\mathcal{S}}(t + 1) &= x^{\mathcal{S}}(t) \end{aligned} \quad (5)$$

It can be proven that Q^{11} is asymptotically stable $((Q^{11})^t \to 0)$. Henceforth, $x^{\mathcal{R}}(t) \to x^{\mathcal{R}}(\infty)$ for $t \to +\infty$ with the limit opinions satisfying the relation

$$x^{\mathcal{R}}(\infty) = Q^{11}x^{\mathcal{R}}(\infty) + Q^{12}x^{\mathcal{S}}(0) \quad (6)$$

If we define $\Xi := (I - Q^{11})^{-1}Q^{12}$, we can write $x^{\mathcal{R}}(\infty) = \Xi x^{\mathcal{S}}(0)$. It is easy to see that Ξ has nonnegative elements and that $\sum_s \Xi_{us} = 1$ for all $u \in \mathcal{R}$: asymptotic opinions of regular agents are thus convex combinations of the opinions of stubborn agents. If all stubborn agents are in the same state x, then, consensus is reached by all agents in the point x. However, typically, consensus is not reached in such a system: we will discuss an example below.

There is a useful alternative interpretation of the asymptotic opinions. Interpreting the graph $\mathcal{G}$ as an electrical circuit where edges are unit resistors, relation (6) can be seen as a Laplace-type equation on the graph $\mathcal{G}$ with boundary conditions given by assigning the voltage $x^{\mathcal{S}}(0)$ to the stubborn agents. In this way, $x^{\mathcal{R}}(\infty)$ can be interpreted as the vector of voltages of the regular agents when stubborn agents have fixed voltage $x^{\mathcal{S}}(0)$. Thanks to the electrical analogy, we can compute the asymptotic opinion of the agents by computing the voltages in the various nodes in the graph. We propose a concrete application in the following example.

Example 5 (Stubborn agents in a line graph) Consider the line graph $L_N = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, \ldots, N\}$ and where $\mathcal{E} = \{(u, u + 1), (u + 1, u) \mid u = 1, \ldots, N - 1\}$. Assume that $\mathcal{S} = \{1, N\}$ and $\mathcal{R} = \mathcal{V} \setminus \mathcal{S}$. Consider the graph as an electrical circuit. Replacing the line with a single edge connecting 1 and N having resistance $N - 1$ and applying Ohm's law, we obtain that the current flowing from 1 to N is equal to $\Phi = (N - 1)^{-1}[x_N^{\mathcal{S}}(0) - x_1^{\mathcal{S}}(0)]$. If we now fix an arbitrary node $v \in \mathcal{V}$ and applying again the same arguments in the part of graph from 1 to v, we obtain that the voltage at v, $x_v^{\mathcal{R}}(\infty)$ satisfies the relation $x_v^{\mathcal{R}}(\infty) - x_1^{\mathcal{S}}(0) = \Phi(v - 1)$. We thus obtain

$$x_v^{\mathcal{R}}(\infty) = x_1^{\mathcal{S}}(0) + \frac{v-1}{N-1}[x_N^{\mathcal{S}}(0) - x_1^{\mathcal{S}}(0)].$$

In Acemoglu et al. (2013), further examples are discussed showing how, because of the topology of the graph, different asymptotic configurations may show up. While in graphs presenting bottlenecks polarization phenomena can be recorded, in graphs where the convergence rate is low, there will be a typical asymptotic opinion shared by most of the regular agents.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Information-Based Multi-Agent Systems

Bibliography

Acemoglu D, Como G, Fagnani F, Ozdaglar A (2013) Opinion fluctuations and disagreement in social networks. Math Oper Res 38(1):1–27
Boyd S, Ghosh A, Prabhakar B, Shah D (2006) Randomized gossip algorithms. IEEE Trans Inf Theory 52(6):2508–2530
Carli R, Fagnani F, Speranzon A, Zampieri S (2008) Communication constraints in the average consensus problem. Automatica 44(3):671–684
Castellano C, Fortunato S, Loreto V (2009) Statistical physics of social dynamics. Rev Modern Phys 81:591–646
Fax JA, Murray RM (2004) Information flow and cooperative control of vehicle formations. IEEE Trans Autom Control 49(9):1465–1476
Galton F (1907) Vox populi. Nature 75:450–451
Gantmacher FR (1959) The theory of matrices. Chelsea Publishers, New York
Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Autom Control 48(6):988–1001
Levin DA, Peres Y, Wilmer EL (2008) Markov chains and mixing times. AMS, Providence
Strogatz SH (2003) Sync: the emerging science of spontaneous order. Hyperion, New York
Surowiecki J (2004) The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. Little, Brown. (Traduzione italiana: *La saggezza della folla*, Fusi Orari, 2007)

Control and Optimization of Batch Processes

Dominique Bonvin
Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

Abstract

A batch process is characterized by the repetition of time-varying operations of finite duration. Due to the repetition, there are two independent "time" variables, namely, the run time during a batch and the batch counter. Accordingly, the control and optimization objectives can be defined for a given batch or over several batches. This entry describes the various control and optimization strategies available for the operation of batch processes. These include conventional feedback control, predictive control, iterative learning control, and run-to-run control on the one hand and model-based repeated optimization and model-free self-optimizing schemes on the other.

Keywords

Batch control; Batch process optimization; Dynamic optimization; Iterative learning control; Run-to-run control; Run-to-run optimization

Introduction

Batch processing is widely used in the manufacturing of goods and commodity products, in particular in the chemical, pharmaceutical, and food industries. These industries account for several billion US dollars in annual sales. Batch operation differs significantly from continuous operation. While in continuous operation the process is maintained at an economically desirable operating point, the process evolves continuously from an initial to a final time in batch processing. In the chemical industry, for example, since the design of a continuous plant requires substantial engineering effort, continuous operation is rarely

Implementation aspect	Control objectives	
	Run-time references $y_{ref}(t)$ or $y_{ref}[0,t_f]$	Run-end references z_{ref}
Online (within-run)	① Feedback control $u_k(t) \rightarrow y_k(t) \rightarrow y_k[0,t_f]$ ⬆ FBC	② Predictive control $\dot{u}_k(t) \rightarrow z_{pred,k}(t)$ ⬆ MPC
Iterative (run-to-run)	③ Iterative learning control $u_k[0,t_f] \rightarrow y_k[0,t_f]$ ⬆ ILC with run delay	④ Run-to-run control $\mathcal{U}(\pi_k) = u_k[0,t_f] \rightarrow z_k$ ⬆ R2R with run delay

Control and Optimization of Batch Processes, Fig. 1
Control strategies for batch processes. The strategies are classified according to the control objectives (horizontal division) and the implementation aspect (vertical division). Each objective can be met either online or iteratively over several batches depending on the type of measurements available. u_k represents the input vector for the kth batch, $u_k[0,t_f]$ the corresponding input trajectories, $y_k(t)$ the run-time outputs measured online, and z_k the run-end outputs available at the final time. FBC stands for "feedback control," MPC for "model predictive control," ILC for "iterative learning control," and R2R for "run-to-run control"

used for low-volume production. Discontinuous operations can be of the batch or semi-batch type. In batch operations, the products to be processed are loaded in a vessel and processed without material addition or removal. This operation permits more flexibility than continuous operation by allowing adjustment of the operating conditions and the final time. Additional flexibility is available in semi-batch operations, where products are continuously added by adjusting the feed rate profile. We use the term batch process to include semi-batch processes.

Batch processes dealing with reaction and separation operations include reaction, distillation, absorption, extraction, adsorption, chromatography, crystallization, drying, filtration, and centrifugation. The operation of batch processes involves recipes developed in the laboratory. A sequence of operations is performed in a prespecified order in specialized process equipment, yielding a fixed amount of product. The sequence of tasks to be carried out on each piece of equipment, such as heating, cooling, reaction, distillation, crystallization, and drying, is predefined. The desired production volume is then achieved by

repeating the processing steps on a predetermined schedule.

The main characteristics of batch process operations include the absence of steady state, the presence of constraints, and the repetitive nature. These characteristics bring both challenges and opportunities to the operation of batch processes (Bonvin 1998). The challenges are related to the fact that the available models are often poor and incomplete, especially since they need to represent a wider range of operating conditions than in the case of continuous processes. Furthermore, although product quality must be controlled, this variable is usually not available online but only at run end. On the other hand, opportunities stem from the fact that industrial chemical processes are often slow, which facilitates larger sampling periods and extensive online computations. In addition, the repetitive nature of batch processes opens the way to run-to-run process improvement (Bonvin et al. 2006). More information on batch processes and their operation can be found in Seborg et al. (2004) and Nagy and Braatz (2003). Next, we will successively address the control and the optimization of batch processes.

Control of Batch Processes

Control of batch processes differs from control of continuous processes in two main ways. First, since batch processes have no steady-state operating point, at least some of the set points are time-varying profiles. Second, batch processes are repeated over time and are characterized by two independent variables, the run time t and the run counter k. The independent variable k provides additional degrees of freedom for meeting the control objectives when these objectives do not necessarily have to be completed in a single batch but can be distributed over several successive batches. This situation brings into focus the concept of run-end outputs, which need to be controlled but are only available at the completion of the batch. The most common run-end output is product quality. Consequently, the control of batch processes encompasses four different strategies (Fig. 1):

1. *Online control of run-time outputs.* This control approach is similar to that used in continuous processing. However, although some controlled variables, such as temperature in isothermal operation, remain constant, the key process characteristics, such as process gain and time constants, can vary considerably because operation occurs along state trajectories rather than at a steady-state operating point. Hence, adaptation in run time t is needed to handle the expected variations. Feedback control is implemented using PID techniques or more sophisticated alternatives (Seborg et al. 2004).

2. *Online control of run-end outputs.* In this case it is necessary to predict the run-end outputs z based on measurements of the run-time outputs y. Model predictive control (MPC) is well suited to this task (Nagy and Braatz 2003). However, the process models available for prediction are often simplified and thus of limited accuracy.

3. *Iterative control of run-time outputs.* The manipulated variable profiles can be generated using iterative learning control (ILC), which exploits information from previous runs

(Moore 1993). This strategy exhibits the limitations of open-loop control with respect to the current run, in particular the fact that there is no feedback correction for run-time disturbances. Nevertheless, this scheme is useful for generating a time-varying feedforward input term.

4. *Iterative control of run-end outputs.* In this case the input profiles are parameterized as $u_k[0, t_f] = \mathcal{U}(\pi_k)$ using the input parameters π_k. The batch process is thus seen as a static map between the input parameters π_k and the run-end outputs z_k (Francois et al. 2005).

It is also possible to combine online and run-to-run control for both y and z. However, in such a combined scheme, care must be taken so that the online and run-to-run corrective actions do not oppose each other. Stability during run time and convergence in run index must be guaranteed (Srinivasan and Bonvin 2007a).

Optimization of Batch Processes

The process variables undergo significant changes during batch operation. Hence, the major objective in batch operations is not to keep the system at optimal constant set points but rather to determine input profiles that optimize an objective function expressing the system performance.

Problem Formulation

A typical optimization problem in the context of batch processes is

$$\min_{u_k[0,t_f]} J_k = \phi\big(x_k(t_f)\big)$$
$$+ \int_0^{t_f} L\big(x_k(t), u_k(t), t\big)\, dt \quad (1)$$

subject to

$$\dot{x}_k(t) = F\big(x_k(t), u_k(t)\big), \quad x_k(0) = x_{k,0} \quad (2)$$
$$S\big(x_k(t), u_k(t)\big) \le 0, \quad T\big(x_k(t_f)\big) \le 0, \quad (3)$$

where x represents the state vector, J the scalar cost to be minimized, S the run-time constraints, T the run-end constraints, and t_f the final time.

In constrained optimal control problems, the solution often lies on the boundary of the feasible region. Batch processes involve run-time constraints on inputs and states as well as run-end constraints.

Optimization Strategies

As can be seen from the cost objective (1), optimization requires information about the complete run and thus cannot be implemented in real time using only online measurements. Some information regarding the future of the run is needed in the form of either a process model capable of prediction or measurements from previous runs. Accordingly, measurement-based optimization methods can be classified depending on whether or not a process model is used explicitly for implementation, as illustrated in Fig. 2 and discussed next:

1. *Online explicit optimization.* This approach is similar to model predictive control (Nagy and Braatz 2003). Optimization uses a process model explicitly and is repeated whenever a new set of measurements becomes available. This scheme involves two steps, namely,

updating the initial conditions for the subsequent optimization (and optionally the parameters of the process model) and numerical optimization based on the updated process model (Abel et al. 2000). Since both steps are repeated as measurements become available, the procedure is also referred to as repeated online optimization. The weakness of this method is its reliance on the model; if the model is not updated, its accuracy plays a crucial role. However, when the model is updated, there is a conflict between parameter identification and optimization since parameter identification requires persistency of excitation, that is, the inputs must be sufficiently varied to uncover the unknown parameters, a condition that is usually not satisfied when near-optimal inputs are applied. Note that, instead of computing the input $u_k^*[t, t_f]$, it is also possible to use a receding horizon and compute only $u_k^*[t, t + T]$, with T the control horizon (Abel et al. 2000).

2. *Online implicit optimization.* In this scenario, measurements are used to update the inputs directly, that is, without the intermediary of a process model. Two classes of techniques can be identified. In the first class, an update law that approximates the optimal solution

Implementation aspect	Use of process model	
	Explicit optimization (with process model)	Implicit optimization (without process model)
Online (within-run)	① *Repeated online optimization* $y_k[0,t] \xrightarrow{\text{EST}} \hat{x}_k(t) \xrightarrow{\text{OPT}} u_k^*[t,t_f]$ ⇡ repeat online	② *Online input update using measurements* $y_k(t) \xrightarrow{\text{Approx. of opt. solution}} u_k^*(t)$ $y_k[0,t] \xrightarrow{\text{NCO prediction}} NCO \longrightarrow u_k^*(t)$
Iterative (run-to-run)	③ *Repeated run-to-run optimization* $y_k[0,t_f] \xrightarrow{\text{IDENT}} \hat{\theta}_k \xrightarrow{\text{OPT}} u_{k+1}^*[0,t_f]$ ⇡ repeat with run delay	④ *Run-to-run input update using measurements* $y_k[0,t_f] \xrightarrow{\text{NCO evaluation}} NCO \longrightarrow u_{k+1}^*[0,t_f]$ ⇡ repeat with run delay

Control and Optimization of Batch Processes, Fig. 2 Optimization strategies for batch processes. The strategies are classified according to whether or not a process model is used for implementation (horizontal division). Furthermore, each class can be implemented either online

or iteratively over several runs (vertical division). EST stands for "estimation," IDENT for "identification," OPT for "optimization," and NCO for "necessary conditions of optimality"

is sought. For example, a neural network is trained with data corresponding to optimal behavior for various uncertainty realizations and used to update the inputs (Rahman and Palanki 1996). The second class of techniques relies on transforming the optimization problem into a control problem that enforces the necessary conditions of optimality (NCO) (Srinivasan and Bonvin 2007b). The NCO involve constraints that need to be made active and sensitivities that need to be pushed to zero. Since some of these NCO are evaluated at run time and others at run end, the control problem involves both run-time and run-end outputs. The main issue is the measurement or estimation of the controlled variables, that is, the constraints and sensitivities that constitute the NCO.

3. *Iterative explicit optimization.* The steps followed in run-to-run explicit optimization are the same as in online explicit optimization. However, there is substantially more data available at the end of the run as well as sufficient computational time to refine the model by updating its parameters and, if needed, its structure. Furthermore, data from previous runs can be collected for model update (Rastogi et al. 1992). As with online explicit optimization, this approach suffers from the conflict between estimation and optimization.

4. *Iterative implicit optimization.* In this scenario, the optimization problem is transformed into a control problem, for which the control approaches in the second row of Fig. 1 are used to meet the run-time and run-end objectives (Francois et al. 2005). The approach, which is conceptually simple, might be experimentally expensive since it relies more on data.

These complementary measurement-based optimization strategies can be combined by implementing some aspects of the optimization online and others on a run-to-run basis. For instance, in explicit schemes, the states can be estimated online, while the model parameters can be estimated on a run-to-run basis. Similarly, in implicit optimization, approximate update

laws can be implemented online, leaving the responsibility for satisfying terminal constraints and sensitivities to run-to-run controllers.

Summary and Future Directions

Batch processing presents several challenges. Since there is little time for developing appropriate dynamic models, there is a need for improved data-driven control and optimization approaches. These approaches require the availability of online concentration-specific measurements such as chromatographic and spectroscopic sensors, which are not yet readily available in production.

Technically, the main operational difficulty in batch process improvement lies in the presence of run-end outputs such as final quality, which cannot be measured during the run. Although model-based solutions are available, process models in the batch area tend to be poor. On the other hand, measurement-based optimization for a given batch faces the challenge of having to know about the future to act during the batch. Consequently, the main research push is in the area of measurement-based optimization and the use of data from both the current and previous batches for control and optimization purposes.

Cross-References

▶ Industrial MPC of continuous processes
▶ Iterative Learning Control
▶ Multiscale Multivariate Statistical Process Control
▶ Scheduling of Batch Plants
▶ State Estimation for Batch Processes

Bibliography

Abel O, Helbig A, Marquardt W, Zwick H, Daszkowski T (2000) Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. J Process Control 10(4):351–362
Bonvin D (1998) Optimal operation of batch reactors – a personal view. J Process Control 8(5–6):355–368

Bonvin D, Srinivasan B, Hunkeler D (2006) Control and optimization of batch processes: improvement of process operation in the production of specialty chemicals. IEEE Control Syst Mag 26(6): 34–45

Francois G, Srinivasan B, Bonvin D (2005) Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. J Process Control 15(6):701–712

Moore KL (1993) Iterative learning control for deterministic systems. Advances in industrial control. Springer, London

Nagy ZK, Braatz RD (2003) Robust nonlinear model predictive control of batch processes. AIChE J 49(7):1776–1786

Rahman S, Palanki S (1996) State feedback synthesis for on-line optimization in the presence of measurable disturbances. AIChE J 42:2869–2882

Rastogi A, Fotopoulos J, Georgakis C, Stenger HG (1992) The identification of kinetic expressions and the evolutionary optimization of specialty chemical batch reactors using tendency models. Chem Eng Sci 47(9–11):2487–2492

Seborg DE, Edgar TF, Mellichamp DA (2004) Process dynamics and control. Wiley, New York

Srinivasan B, Bonvin D (2007a) Controllability and stability of repetitive batch processes. J Process Control 17(3):285–295

Srinivasan B, Bonvin D (2007b) Real-time optimization of batch processes by tracking the necessary conditions of optimality. Ind Eng Chem Res 46(2):492–504

Control Applications in Audio Reproduction

Yutaka Yamamoto
Department of Applied Analysis and Complex Dynamical Systems, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Abstract

This entry gives a brief overview of the recent developments in audio sound reproduction via modern sampled-data control theory. We first review basics in the current sound processing technology and then proceed to the new idea derived from sampled-data control theory, which is different from the conventional Shannon paradigm based on the perfect band-limiting hypothesis. The hybrid nature of sampled-data systems provides an optimal platform for dealing with signal

processing where the ultimate objective is to reconstruct the original analog signal one started with. After discussing some fundamental problems in the Shannon paradigm, we give our basic problem formulation that can be solved using modern sampled-data control theory. Examples are given to illustrate the results.

Keywords

Introduction: Status Quo

Consider the problem of reproducing sounds from recorded media such as compact discs. The current CD format is recorded at the sampling frequency 44.1 kHz. It is commonly claimed that the highest frequency for human audibility is 20 kHz, whereas the upper bound of reproduction in this format is believed to be the half of 44.1 kHz, i.e., 22.1 kHz, and hence, this format should have about 10 % margin against the alleged audible limit of 20 kHz.

CD players of early days used to process such digital signals with the simple zero-order hold at this frequency, followed by an analog low-pass filter. This process requires a sharp low-pass characteristic to cut out unnecessary high frequency beyond 20 kHz. However, a sharp cutoff low-pass characteristic inevitably requires a high-order filter which in turn introduces a large amount of phase shift distortion around the cutoff frequency.

To circumvent this defect, there was introduced the idea of oversampling DA converter that is realized by the combination of a digital filter and a low-order analog filter (Zelniker and Taylor 1994). This is based on the following principle:

Let $\{f(nh)\}_{n=-\infty}^{\infty}$ be a discrete-time signal obtained from a continuous-time signal $f(\cdot)$ by sampling it with sampling period h. The *upsampler* appends the value 0, $M - 1$ times, between two adjacent sampling points:

**Control Applications in
Audio Reproduction,
Fig. 1** Upsampler for
$M = 2$
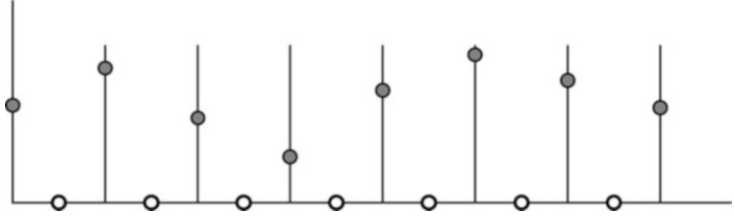
$$(\uparrow Mw)[k] := \begin{cases} w(\ell), & k = M\ell \\ 0, & \text{elsewhere.} \end{cases} \quad (1)$$

See Fig. 1 for the case $M = 2$. This has the effect of making the unit operational time M times faster.

The bandwidth will also be expanded by M times and the *Nyquist frequency* (i.e., half the sampling frequency) becomes $M\pi/h$ [rad/sec]. As we see in the next section, the Nyquist frequency is often regarded as the true bandwidth of the discrete-time signal $\{f(nh)\}_{n=-\infty}^{\infty}$. But this upsampling process just insert zeros between sampling points, and the real information contents (the true bandwidth) is not really expanded. As a result, the copy of the frequency content for $[0, \pi/h)$ appears as a mirror image repeatedly over the frequency range above π/h. This distortion is called *imaging*. In order to avoid the effect of such mirrored frequency components, one often truncates the frequency components beyond the (original) Nyquist frequency via a digital low-pass filter that has a sharp roll-off characteristic. One can then complete the digital to analog (DA) conversion process by postposing a slowly decaying analog filter. This is the idea of an *oversampling DA converter* (Zelniker and Taylor 1994). The advantage here is that by allowing a much wider frequency range, the final analog filter can be a low-order filter and hence yields a relatively small amount of phase distortion supported in part by the linear-phase characteristic endowed on the digital filter preceding it.

Signal Reconstruction Problem

As before, consider the sampled discrete-time signal $\{f(nh)\}_{n=-\infty}^{\infty}$ obtained from a continuous-time signal f. The main question is how we can recover the original continuous-time signal $f(\cdot)$ from sampled data. This is clearly an ill-posed problem without any assumption on f because there are infinitely many functions that can match the sampled data $\{f(nh)\}_{n=-\infty}^{\infty}$. Hence, one has to impose a reasonable a priori assumption on f to sensibly discuss this problem.

The following sampling theorem gives one answer to this question:

Theorem 1 *Suppose that the signal $f \in L^2$ is perfectly band-limited, in the sense that there exists $\omega_0 \leq \pi/h$ such that the Fourier transform $\hat{f}$ of f satisfies*

$$\hat{f}(\omega) = 0, \quad |\omega| \geq \omega_0,. \quad (2)$$

Then

$$f(t) = \sum_{n=-\infty}^{\infty} f(nh)\frac{\sin \pi(t/h - n)}{\pi(t/h - n)}. \quad (3)$$

This theorem states that if the signal f does not contain any high-frequency components beyond the *Nyquist frequency* π/h, then the original signal f can be uniquely reconstructed from its sampled-data $\{f(nh)\}_{n=-\infty}^{\infty}$. On the other hand, if this assumption does not hold, then the result does not necessarily hold. This is easy to see via a schematic representation in Fig. 2.

If we sample the sinusoid in the upper figure in Fig. 2, these sampled values would turn out to be compatible with another sinusoid with much lower frequency as the lower figure shows. In other words, this sampling period does not have enough resolution to distinguish these two sinusoids. The maximum frequency below where there does not occur such a phenomenon is the Nyquist frequency. The sampling theorem above asserts that it is half of the sampling frequency $2\pi/h$, that is, π/h [rad/sec]. In other words, if
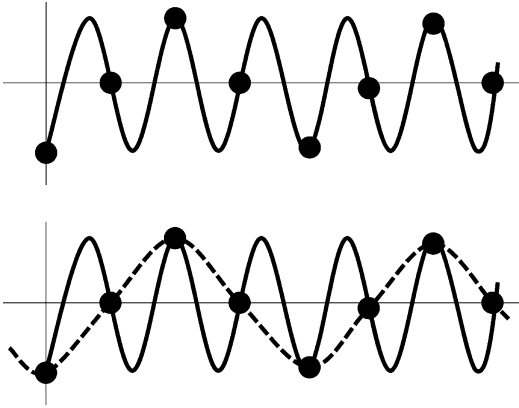
Control Applications in Audio Reproduction, Fig. 2
Aliasing

we can assume that the original signal contains no frequency components beyond the Nyquist frequency, then one can uniquely reconstruct the original analog signal f from its sampled-data $\{f(nh)\}_{n=-\infty}^{\infty}$. On the other hand, if this assumption does not hold, the distortion depicted in Fig. 2 occurs; this is called *aliasing*.

This is the content of the sampling theorem. It has been widely accepted as the basis for digital signal processing that bridges analog to digital. Concrete applications such as CD, MP3, or images are based on this principle in one way or another.

Difficulties

However, this paradigm (hereafter the *Shannon paradigm*) of the perfect band-limiting hypothesis and the resulting sampling theorem renders several difficulties as follows:

- The reconstruction formula (3) is not causal, i.e., one needs future sampled values to reconstruct the present value $f(t)$. One can remedy this defect by allowing a certain amount of delay in reconstruction, but this delay can depend on how fast the formula converges.
- This formula is known to decay slowly; that is, we need many terms to approximate if we use this formula as it is.
- The perfect band-limiting hypothesis is hardly satisfied in reality. For example, for CDs, the

Nyquist frequency is 22.05 kHz, and the energy distribution of real sounds often extends way over 20 kHz.
- To remedy this, one often introduces a band-limiting low-pass filter, but it can introduce distortions due to the Gibbs phenomenon, due to a required sharp decay in the frequency domain. See Fig. 3.

This is the Gibbs phenomenon well known in Fourier analysis. A sharp truncation in the frequency domain yields such a ringing effect.

In view of such drawbacks, there has been revived interest in the extension of the sampling theorem in various forms since the 1990s. There is by now a stream of papers that aim at studying signal reconstruction under the assumption of nonideal signal acquisition devices; an excellent survey is given in Unser (2000). In this research framework, the incoming signal is supposed to be acquired through a nonideal analog filter (acquisition device) and sampled, and then the reconstruction process attempts to recover the original signal. The idea is to place the problem into the framework of the (orthogonal or oblique) projection theorem in a Hilbert space (usually L^2) and then project the signal space to the subspace generated by the shifted reconstruction functions. It is often required that the process give a *consistent* result, i.e., if we subject the reconstructed signal to the whole process again, it should yield the same sampled values from which it was reconstructed (Unser and Aldroubi 1994).

In what follows, we take a similar viewpoint, that is, the incoming signals are acquired through a nonideal filter, but develop a methodology different from the projection method, relying on sampled-data control theory.
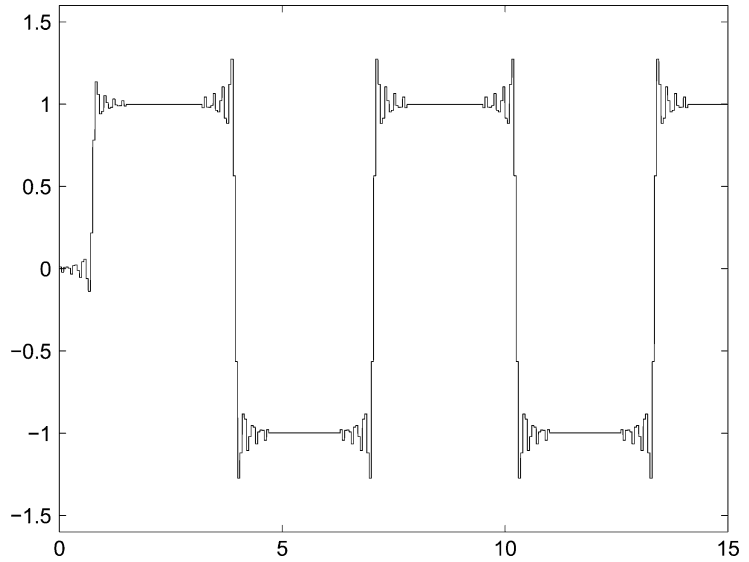
The Signal Class

We have seen that the perfect band-limiting hypothesis is restrictive. Even if we adopt it, it is a fairly crude model for analog signals to allow for a more elaborate study.

Let us now pose the question: *What class of functions should we process in such systems?*

Control Applications in Audio Reproduction, Fig. 3 Ringing due to the Gibbs phenomenon
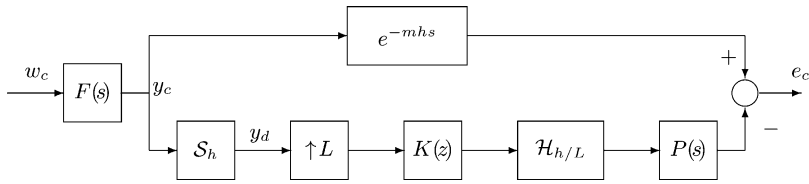
Control Applications in Audio Reproduction, Fig. 4 Error system for sampled-data design

Consider the situation where one plays a musical instrument, say, a guitar. A guitar naturally has a frequency characteristic. When one picks a string, it produces a certain tone along with its harmonics, as well as a characteristic transient response. All these are governed by a certain frequency decay curve, demanded by the physical characteristics of the guitar. Let us suppose that such a frequency decay is governed by a rational transfer function $F(s)$, and it is driven by varied exogenous inputs.

Consider Fig. 4. The exogenous analog signal $w_c \in L^2$ is applied to the analog filter $F(s)$. This $F(s)$ is not an ideal filter and hence its bandwidth is not limited below the Nyquist frequency. The signal w_c drives $F(s)$ to produce the target analog signal y_c, which should be the signal to be reconstructed. It is then sampled by sampler $\mathcal{S}_h$ and becomes the recorded or transmitted digital signal y_d. The objective here is to reconstruct the target analog signal y_c out of this sampled signal y_d. In order to recover

the frequency components beyond the Nyquist frequency, one needs a faster sampling period, so we insert the upsampler $\uparrow L$ to make the sampling period h/L. This upsampled signal is processed by digital filter $K(z)$ and then becomes a continuous-time signal again by going through the hold device $\mathcal{H}_{h/L}$. It will then be processed by analog filter $P(s)$ to be smoothed out. The obtained signal is then compared with delayed analog signal $y_c(t - mh)$ to form the delayed error signal e_c. The objective is then to make this error e_c as small as possible. The reason for allowing delay e^{-mhs} is to accommodate certain processing delays. This is the idea of the block diagram Fig. 4.

The performance index we minimize is the induced norm of the transfer operator T_{ew} from w_c to e_c:

$$\|T_{ew}\|_\infty := \sup_{w_c \neq 0} \frac{\|e_c\|_2}{\|w_c\|_2}. \qquad (4)$$

In other words, the H^∞-norm of the sampled-data control system Fig. 4. Our objective is then to solve the following problem:

Filter Design Problem

Given the system specified by Fig. 4. For a given performance level $\gamma > 0$, find a filter $K(z)$ such that

$$\|T_{ew}\|_\infty < \gamma.$$

This is a sampled-data H^∞ (sub-)optimal control problem. This can be solved by using the standard solution method for sampled-data control systems (Chen and Francis 1995a; Yamamoto 1999; Yamamoto et al. 2012). The only anomaly here is that the system in Fig. 4 contains a delay element e^{-mhs} which is infinite dimensional. However, by suitably approximating this delay by successive series of shift registers, one can convert the problem to an appropriate finite-dimensional discrete-time problem (Yamamoto et al. 1999, 2002, 2012).

This problem setting has the following features:

1. One can optimize the continuous-time performance under the constraint of discrete-time filters.
2. By setting the class of input functions as L^2 functions band-limited by $F(s)$, one can capture the continuous-time error signal e_c and its worst-case norm in the sense of (4).

The first feature is due to the advantage of sampled-data control theory. It is a great advantage of sampled-data control theory that allows the mixture of continuous- and discrete-time components. This is in marked contrast to the Shannon paradigm where continuous-time performance is really demanded by the artificial perfect band-limiting hypothesis.

The second feature is an advantage due to H^∞ control theory. Naturally, we cannot have an access to each *individual* error signal e_c, but we can still control the *overall performance* from w_c to e_c in terms of the H^∞ norm that guarantees the worst-case performance. This is in clear contrast with the classical case where only a representative response, e.g., impulse response

in the case of H^2, is targeted. Furthermore, since we can control the continuous-time performance of the worst-case error signal, the present method can indeed minimize (continuous-time) phase errors. This is an advantage usually not possible with conventional methods since they mainly discuss the gain characteristics of the designed filters only. By the very property of minimizing the H^∞ norm of the *continuous-time error signal e_c*, the present method can even control the phase errors and yield much less phase distortion even around the cutoff frequency.

Figure 5 shows the response of the proposed sampled-data filter against a rectangular wave, with a suitable first- or second-order analog filter $F(s)$; see Yamamoto et al. (2012) for more details. Unlike Fig. 3, the overshoot is controlled to be minimum.

The present method has been patented (Fujiyama et al. 2008; Yamamoto 2006; Yamamoto and Nagahara 2006) and implemented into sound processing LSI chips as a core technology by Sanyo Semiconductors and successfully used in mobile phones, digital voice recorders, and MP3 players; their cumulative production has exceeded 40 million units as of the end of 2012.

Summary and Future Directions

We have presented basic ideas of new signal processing theory derived from sampled-data control theory. The theory has the advantage that is not possible with the conventional projection methods, whether based on the perfect band-limiting hypothesis or not.

The application of sampled-data control theory to digital signal processing was first made by Chen and Francis (1995b) with performance measure in the discrete-time domain; see also Hassibi et al. (2006). The present author and his group have pursued the idea presented in this entry since 1996 (Khargonekar and Yamamoto 1996). See Yamamoto et al. (2012) and references therein. For the background of sampled-data

Control Applications in Audio Reproduction, Fig. 5 Response of the proposed sampled-data filter against a rectangular wave
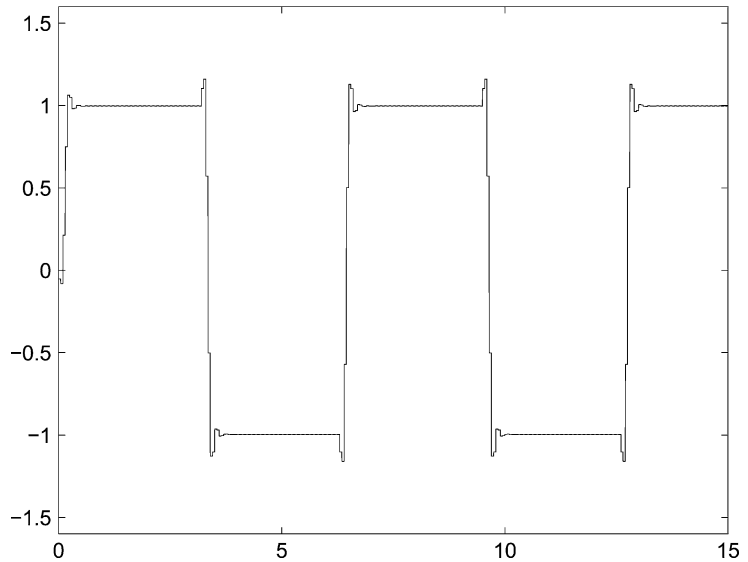
control theory, consult, e.g., Chen and Francis (1995a) and Yamamoto (1999).

The same philosophy of emphasizing the importance of analog performance was proposed and pursued recently by Unser and co-workers (1994), Unser (2005), and Eldar and Dvorkind (2006). The crucial difference is that they rely on L^2/H^2 type optimization and orthogonal or oblique projections, which are very different from our method here. In particular, such projection methods can behave poorly for signals outside the projected space. The response shown in Fig. 3 is a typical such example.

Applications to image processing is discussed in Yamamoto et al. (2012). An application to Delta-Sigma DA converters is studied in Nagahara and Yamamoto (2012). Again, the crux of the idea is to assume a signal generator model and then design an optimal filter in the sense of Fig. 4 or a similar diagram with the same idea. This idea should be applicable to a much wider class of problems in signal processing and should prove to have more impact.

Some processed examples of still and moving images are downloadable from the site: http://www-ics.acs.i.kyoto-u.ac.jp/~yy/

For sampling theorem, see Shannon (1949), Unser (2000), and Zayed (1996), for example. Note, however, that Shannon himself (1949) did not claim originality on this theorem; hence, it is misleading to attribute this theorem solely to Shannon. See Unser (2000) and Zayed (1996) for some historical accounts. For a general background in signal processing, Vetterli et al. (2013) is useful.

Cross-References

▸ H-Infinity Control
▸ Optimal Sampled-Data Control
▸ Sampled-Data Systems

Bibliography

Chen T, Francis BA (1995a) Optimal sampled-data control systems. Springer, New York
Chen T, Francis BA (1995b) Design of multirate filter banks by $\mathcal{H}_\infty$ optimization. IEEE Trans Signal Process 43:2822–2830

Eldar YC, Dvorkind TG (2006) A minimum squared-error framework for generalized sampling. IEEE Trans Signal Process 54(6):2155–2167

Fujiyama K, Iwasaki N, Hirasawa Y, Yamamoto Y (2008) High frequency compensator and reproducing device. US patent 7,324,024 B2, 2008

Hassibi B, Erdogan AT, Kailath T (2006) MIMO linear equalization with an H^∞ criterion. IEEE Trans Signal Process 54(2):499–511

Khargonekar PP, Yamamoto Y (1996) Delayed signal reconstruction using sampled-data control. In: Proceedings of 35th IEEE CDC, Kobe, Japan, pp 1259–1263

Nagahara M, Yamamoto Y (2012) Frequency domain min-max optimization of noise-shaping delta-sigma modulators. IEEE Trans Signal Process 60(6):2828–2839

Shannon CE (1949) Communication in the presence of noise. Proc IRE 37(1):10–21

Unser M (2000) Sampling – 50 years after Shannon. Proc IEEE 88(4):569–587

Unser M (2005) Cardinal exponential splines: part II – think analog, act digital. IEEE Trans Signal Process 53(4):1439–1449

Unser M, Aldroubi A (1994) A general sampling theory for nonideal acquisition devices. IEEE Trans Signal Process 42(11):2915–2925

Vetterli M, Kovacčević J, Goyal V (2013) Foundations of signal processing. Cambridge University Press, Cambridge

Yamamoto Y (1999) Digital control. In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering, vol 5. Wiley, New York, pp 445–457

Yamamoto Y (2006) Digital/analog converters and a design method for the pertinent filters. Japanese patent 3,820,331, 2006

Yamamoto Y (2007) New developments in signal processing via sampled-data control theory–continuous-time performance and optimal design. Meas Control (Jpn) 46:199–205

Yamamoto Y, Nagahara M (2006) Sample-rate converters. Japanese patent 3,851,757, 2006

Yamamoto Y, Madievski AG, Anderson BDO (1999) Approximation of frequency response for sampled-data control systems. Automatica 35(4):729–734

Yamamoto Y, Anderson BDO, Nagahara M (2002) Approximating sampled-data systems with applications to digital redesign. In: Proceedings of the 41st IEEE CDC, Las Vegas, pp 3724–3729

Yamamoto Y, Nagahara M, Khargonekar PP (2012) Signal reconstruction via H^∞ sampled-data control theory–beyond the Shannon paradigm. IEEE Trans Signal Process 60:613–625

Zayed AI (1996) Advances in Shannon's sampling theory. CRC Press, Boca Raton

Zelniker G, Taylor FJ (1994) Advanced digital signal processing: theory and applications. Marcel Dekker, New York

Control for High-Speed Nanopositioning

S.O. Reza Moheimani
School of Electrical Engineering & Computer Science, The University of Newcastle, Callaghan, NSW, Australia

Abstract

Over the last two and a half decades we have observed astonishing progress in the field of nanotechnology. This progress is largely due to the invention of Scanning Tunneling Microscope (STM) and Atomic Force Microscope (AFM) in the 1980s. Central to the operation of AFM and STM is a nanopositioning system that moves a sample or a probe, with extremely high precision, up to a fraction of an Angstrom, in certain applications. This note concentrates on the fundamental role of feedback, and the need for model-based control design methods in improving accuracy and speed of operation of nanopositioning systems.

Keywords

Atomic force microscopy; High-precision mechatronic systems; Nanopositioining; Scanning probe microscopy

Introduction

Controlling motion of an actuator to within a single atom, known as nanopositioning, may seem as an impossible task. Yet, it has become a key requirement in many systems to emerge in recent years. In scanning probe microscopy nanopositioning is needed to scan a probe over a sample surface for imaging and to control the interaction between the probe and the surface during interrogation and manipulation (Meyer et al. 2004). Nanopositioning is the enabling technology for mask-less lithography tools under development to replace optical lithography systems (Vettiger et al. 2002). Novel nanopositioning

tools are required for positioning of wafers and for mask alignment in the semiconductor industry (Verma et al. 2005). Nanopositioning systems are vital in molecular biology for imaging, alignment, and nanomanipulation in applications such as DNA analysis (Meldrum et al. 2001) and nanoassembly (Whitesides and Christopher Love 2001). Nanopositioning is an important technology in optical alignment systems (Krogmann 1999). In data storage systems, nanometer-scale precision is needed for emerging probe-storage devices, for dual-stage hard-disk drives, and for next generation tape drives (Cherubini et al. 2012).

The Need for High-Speed Nanopositioning

In all applications of nanopositioning, there is a significant and growing demand for high speeds. The ability to operate a nanopositioner at a bandwidth of tens of kHz, as opposed to today's hundreds of Hz, is the key to unlocking countless technological possibilities in the future (Gao et al. 2000; Pantazi et al. 2008; Salapaka 2003; Sebastian et al. 2008b; Yong et al. 2012). The atomic force microscope (AFM) is an example of such technologies. A typical commercial atomic force microscope is a slow device, taking up to a minute or longer to generate an image. Such imaging speeds are too slow to investigate phenomena with fast dynamics. For example, rapid biological processes that occur in seconds, such as rapid movement of cells or fast dehydration and denaturation of collagen, are too fast to be observed by a typical commercial AFM (Zou et al. 2004). A key obstacle in realizing high-speed and video-rate atomic force microscopy is the limited speed of nanopositioners.

The Vital Role of Feedback Control in High-Speed Nanopositioning

The systems described above depend on a precision mechatronic device, known as a *nanopositioner*, or a *scanner* for their operation.
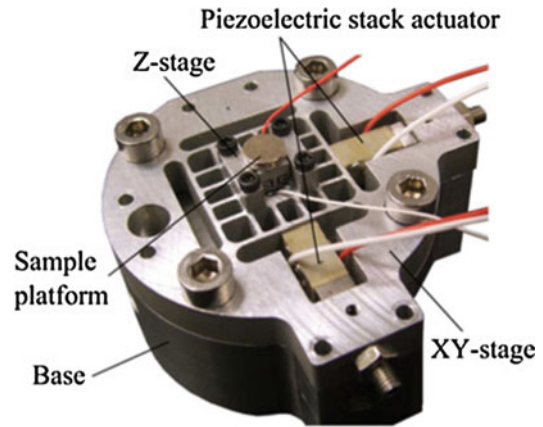
Control for High-Speed Nanopositioning, Fig. 1 A 3DoF flexure-guided high-speed nanopositioner (Yong et al. 2013). The three axes are actuated independently using piezoelectric stack actuators. Movement of lateral axes is measured using capacitive sensors

A high-speed scanner is shown in Fig. 1. In all applications where nanopositioning is a necessity, the key objective is to make the scanner follow, or track, a given reference trajectory (Devasia et al. 2007). A large number of control design methods have been proposed for this purpose, including feedforward control (Clayton et al. 2009), feedback control (Salapaka 2003), and combinations of those (Yong et al. 2009). These control techniques are required in order to compensate for the mechanical resonances of the scanner as well as for various nonlinearities and uncertainties in the dynamics of the nanopositioner. At low speeds, feedforward techniques are usually sufficient to address many of the arising challenges. However, over a wide bandwidth, model uncertainties, sensor noise, and mechanical cross-couplings become significant, and hence feedback control becomes essential to achieve the requisite nanoscale accuracy and precision at high speeds (Devasia et al. 2007; Salapaka 2003).

Control Design Challenges

A feedback loop typically encountered in nanopositioning is illustrated in Fig. 2. The purpose of the feedback controller is to control
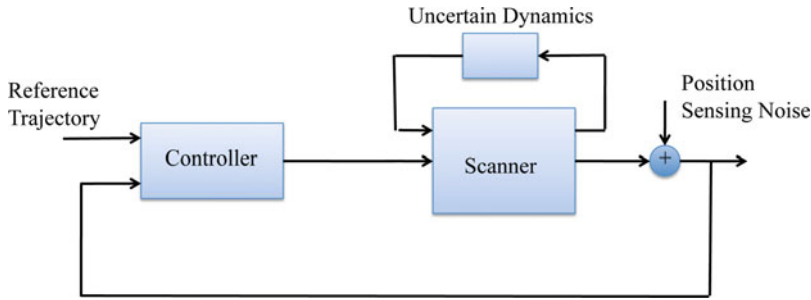
Control for High-Speed Nanopositioning, Fig. 2 A feedback loop typically encountered in nanopositioning. Purpose of the controller is to control the position of the scanner such that it follows the intended reference trajectory based on the position measurement obtained from a position sensor

the position of the scanner such that it follows a given reference trajectory based on the measurement provided by a displacement sensor. The resulting tracking error contains both deterministic and stochastic components. Deterministic errors are typically due to insufficient closed-loop bandwidth. They may also arise from excitation of mechanical resonant modes of the scanner or actuator nonlinearities such as piezoelectric hysteresis and creep (Croft et al. 2001). The factors that limit the achievable closed-loop bandwidth include phase delays and non-minimum phase zeros associated with the actuator and scanner dynamics (Devasia et al. 2007). The dynamics of the nanopositioner, the controller, and the reference trajectory selected for scanning play a key role in minimizing the deterministic component of the tracking error.

Tracking errors of a stochastic nature mostly arise from external noise and vibrations and from position measurement noise. External noise and vibrations can be significantly reduced by operating the nanopositioner in a controlled environment. However, dealing with the measurement noise is a significant challenge (Sebastian et al. 2008a). The feedback loop allows the sensing noise to generate a random positioning error that deteriorates the positioning precision. Increasing the closed-loop bandwidth (to decrease the deterministic errors) tends to worsen this effect. Low sensitivity to measurement noise is, therefore, a key requirement in feedback control design for high-speed nanopositioning and a very hard problem to address.

Summary and Future Directions

While high-precision nanoscale positioning systems have been demonstrated at low speeds, despite an intensive international race spanning several years, the longstanding challenge remains to achieve high-speed motion and positioning with Ångstrom-level accuracy. Overcoming this barrier is believed to be the necessary catalyst for emergence of ground breaking innovations across a wide range of scientific and technological fields. Control is a critical technology to facilitate the emergence of such systems.

Bibliography

Cherubini G, Chung CC, Messner WC, Moheimani SOR (2012) Control methods in data-storage systems. IEEE Trans Control Syst Technol 20(2):296–322

Clayton GM, Tien S, Leang KK, Zou Q, Devasia S (2009) A review of feedforward control approaches in nanopositioning for high-speed SPM. J Dyn Syst Meas Control Trans ASME 131(6):1–19

Croft D, Shed G, Devasia S (2001) Creep, hysteresis, and vibration compensation for piezoactuators: atomic force microscopy application. ASME J Dyn Syst Control 123(1):35–43

Devasia S, Eleftheriou E, Moheimani SOR (2007) A survey of control issues in nanopositioning. IEEE Trans Control Syst Technol 15(5):802–823

Gao W, Hocken RJ, Patten JA, Lovingood J, Lucca DA (2000) Construction and testing of a nanomachining instrument. Precis Eng 24(4):320–328

Krogmann D (1999) Image multiplexing system on the base of piezoelectrically driven silicon microlens arrays. In: Proceedings of the 3rd international conference on micro opto electro mechanical systems (MOEMS), Mainz, pp 178–185

Meldrum DR, Pence WH, Moody SE, Cunningham DL, Holl M, Wiktor PJ, Saini M, Moore MP, Jang L, Kidd M, Fisher C, Cookson A (2001) Automated, integrated modules for fluid handling, thermal cycling and purification of DNA samples for high throughput sequencing and analysis. In: IEEE/ASME international conference on advanced intelligent mechatronics, AIM, Como, vol 2, pp 1211–1219

Meyer E, Hug HJ, Bennewitz R (2004) Scanning probe microscopy. Springer, Heidelberg

Pantazi A, Sebastian A, Antonakopoulos TA, Bachtold P, Bonaccio AR, Bonan J, Cherubini G, Despont M, DiPietro RA, Drechsler U, DurIg U, Gotsmann B, Haberle W, Hagleitner C, Hedrick JL, Jubin D, Knoll A, Lantz MA, Pentarakis J, Pozidis H, Pratt RC, Rothuizen H, Stutz R, Varsamou M, Weismann D, Eleftheriou E (2008) Probe-based ultrahigh-density storage technology. IBM J Res Dev 52(4–5): 493–511

Salapaka S (2003) Control of the nanopositioning devices. In: Proceedings of the IEEE conference on decision and control, Maui

Sebastian A, Pantazi A, Moheimani SOR, Pozidis H, Eleftheriou E (2008a) Achieving sub-nanometer precision in a MEMS storage device during self-servo write process. IEEE Trans Nanotechnol 7(5):586–595. doi:10.1109/TNANO.2008.926441

Sebastian A, Pantazi A, Pozidis H, Eleftheriou E (2008b) Nanopositioning for probe-based data storage [applications of control]. IEEE Control Syst Mag 28(4):26–35

Verma S, Kim W, Shakir H (2005) Multi-axis maglev nanopositioner for precision manufacturing and manipulation applications. IEEE Trans Ind Appl 41(5):1159–1167

Vettiger P, Cross G, Despont M, Drechsler U, Durig U, Gotsmann B, Haberle W, Lantz MA, Rothuizen HE, Stutz R, Binnig GK (2002) The "millipede"-nanotechnology entering data storage. IEEE Trans Nanotechnol 1(1):39–54

Whitesides GM, Christopher Love J (2001) The art of building small. Sci Am 285(3):38–47

Yong YK, Aphale S, Moheimani SOR (2009) Design, identification and control of a flexure-based XY stage for fast nanoscale positioning. IEEE Trans Nanotechnol 8(1):46–54

Yong YK, Moheimani SOR, Kenton BJ, Leang KK (2012) Invited review article: high-speed flexure-guided nanopositioning: mechanical design and control issues. Rev Sci Instrum 83(12):121101

Yong YK, Bhikkaji B, Moheimani SOR (2013) Design, modeling and FPAA-based control of a high-speed atomic force microscope nanopositioner. IEEE/ASME Trans Mechatron 18(3):1060–1071. doi:10.1109/TMECH.2012.2194161

Zou Q, Leang KK, Sadoun E, Reed MJ, Devasia S (2004) Control issues in high-speed AFM for biological applications: collagen imaging example. Asian J Control Spec Issue Adv Nanotechnol Control 6(2): 164–178

Control Hierarchy of Large Processing Plants: An Overview

Cesar de Prada
Departamento de Ingeniería de Sistemas y Automática, University of Valladolid, Valladolid, Spain

Abstract

This entry provides an overview of the so-called control pyramid, which organizes the different types of control tasks in a processing plant in a set of interconnected layers, from basic control and instrumentation to plant-wide economic optimization. These layers have different functions, all of them necessary for the optimal functioning of large processing plants.

Keywords

Control hierarchy; Control pyramid; Model-predictive control; Optimization; Plant-wide control; Real-time optimization

Introduction

Operating a process plant is a complex task involving many different aspects ranging from the control of individual pieces of equipment and of process units to the management of the plant or factory as a whole, including relations with other plants or suppliers.

From the control point of view, the corresponding tasks are traditionally organized in several layers, placing in the bottom the ones closer to the physical processes and in the top those closer to plant-wide management, forming the so-called control pyramid represented in Fig. 1.

The process industry currently faces many challenges, originated from factors such as increased competition among companies and better global market information, new environmental regulations and safety standards, improved quality, or energy efficiency requirements. Many years ago, the main tasks were associated to the
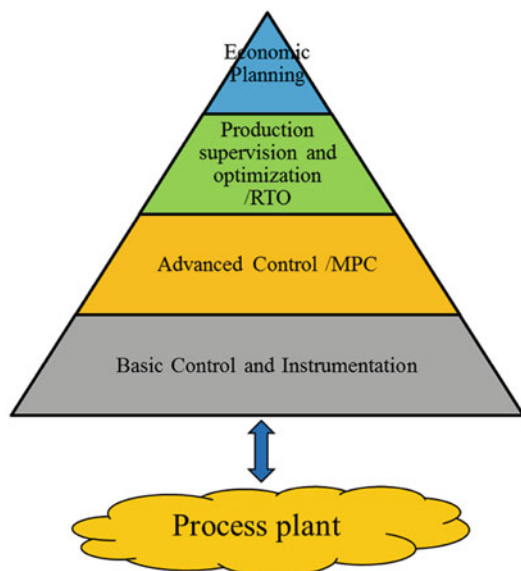
Control Hierarchy of Large Processing Plants: An Overview, Fig. 1 The control pyramid

correct and safe functioning of the individual process units and to the global management of the factory from the point of view of organization and economy. Therefore, only the lower and top layers of the control pyramid were realized by computer-based systems, whereas the intermediate tasks were largely performed by human operators and managers, but more and more the intermediate layers are gaining importance in order to face the abovementioned challenges.

Above the physical plant represented in Fig. 1, there is a layer related to instrumentation and basic control, devoted to obtaining direct process information and maintaining selected process variables close to their desired targets by means of local controllers. Motivated by the need for more efficient operation and better-quality assurance, an improvement of this basic control can be obtained using control structures such as cascades, feed forwards, ratios, and selectors. This is called advanced control in industry, but not in academia, where the word is reserved for more sophisticated controls.

A big step forward took place in the control field with the introduction of model-based predictive control (MBPC/MPC) in the late 1970s

and 1980s, (▶ Industrial MPC of Continuous Processes; Camacho and Bordóns (2004)). MPC aims at regulating a process unit as a whole considering all manipulated and controlled variables simultaneously. It handles all interactions, disturbances, and process constraints using a process model in order to compute the control actions that optimize a control performance index. MPC is built on top of the basic control loops and partly replaces the complex control structures of the advanced control layer adding new functionalities and better control performance. The improvements in control quality and the management of constraints and interactions of the model-predictive controllers open the door for the implementation of local economic optimization. Linked to the MPC controller and taking advantage of its model, an optimizer may look for the best operating point of the unit by computing the controller set points that optimize an economic cost function of the process unit considering the operational constraints of the unit. This task is usually formulated and solved as a linear programming (LP) problem, i.e., based on linear or linearized economic models and cost function (see Fig. 2).

A natural extension of these ideas was to consider the interrelations among the different parts of the processing plants and to look for the steady-state operating point that provides the best economic return and minimum energy expense or optimizes any other economic criterion while satisfying the global production aims and constraints. These optimization tasks are known as real-time optimization (RTO) (▶ Real-Time Optimization of Industrial Processes) and form another layer of the control pyramid.

Finally, when we consider the whole plant operation, obvious links between the RTO and the planning and economic management of the company appear. In particular, the organization and optimization of the flows of raw materials, purchases, etc., involved in the supply chains present important challenges that are placed in the top layer of Fig. 1.

This entry provides an overview of the different layers and associated tasks so that the
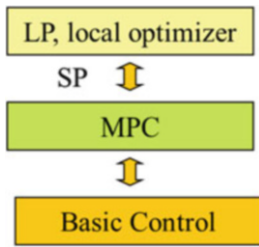
Control Hierarchy of Large Processing Plants: An Overview, Fig. 2 MPC implementation with a local optimizer

reader can place in context the different controllers and related functionalities and tools, as well as appreciate the trends in process control focusing the attention toward the higher levels of the hierarchy and the optimal operation of large-scale processes.

An Alternative View

The implementation in a process factory of the tasks and layers previously mentioned is possible nowadays due to important advances in many fields, such as modeling and identification, control and estimation, optimization methods, and, in particular, software tools, communications, and computing power. Today it is rather common to find in many process plants an information network that follows also a pyramidal structure represented in Fig. 3.

At the bottom, there is the instrumentation layer that includes, besides sensors and actuators connected by the classical analog 4–20 mA signals, possibly enhanced by the transmission of information to and from the sensors by the HART protocol, digital field buses and smart transmitters and actuators that incorporate improved information and intelligence. New functionalities, such as remote calibration, filtering, self-test, and disturbance compensation, provide more accurate measurements that contribute to improving the functioning of local controllers, in the same way as that of new methods and tools available nowadays for instrument monitoring and fault detection and diagnosis. The increased

installation of wireless transmitters and the advances in analytical instrumentation will lead, without doubt, to the development of a stronger information base to support better decisions and operations in the plants.

Information from transmitters is collected in the control rooms that are the core of the second layer. Many of them are equipped with distributed control systems (DCS) that implement monitoring and control tasks. Field signals are received in the control cabinets where a large number of microprocessors execute the data acquisition and regulatory control tasks, sending signals back to the field actuators. Internal buses connect the controllers with the computers that support the displays of the human-machine interface (HMI) for the plant operators of the control room. In the past, DCS were mostly in charge of the regulatory control tasks, including basic control, alarm management, and historians, while interlocking systems related to safety and sequences related to batch operations were implemented either in the DCS or in programmable logic controllers (PLCs): ▶ Programmable Logic Controllers. Today, the bounds are not so clear, due to the increase of the computing power of the PLCs and the added functionalities of the DCS. Safety instrumented systems (SIS) for the maintenance of plant safety are usually implemented in dedicated PLCs, if not hard-wired, but for the rest of the functions, a combination of PLC-like processors with I/O cards and SCADAs (Supervision, Control, And Data Acquisition Systems) is the prevailing architecture. SCADAs act as HMI and information systems collecting large amounts of data that can be used at other levels with different purposes.

Above the basic and advanced control layer, using the information stored in the SCADA as well as other sources, there is an increased number of applications covering diverse fields. Figure 3 depicts the perspective of the computing and information flow architecture and includes a level called supervisory control, placed in direct connection with the control room and the production tasks. It includes, for instance, MPC with local optimizers, statistical process control (SPC) for quality and production
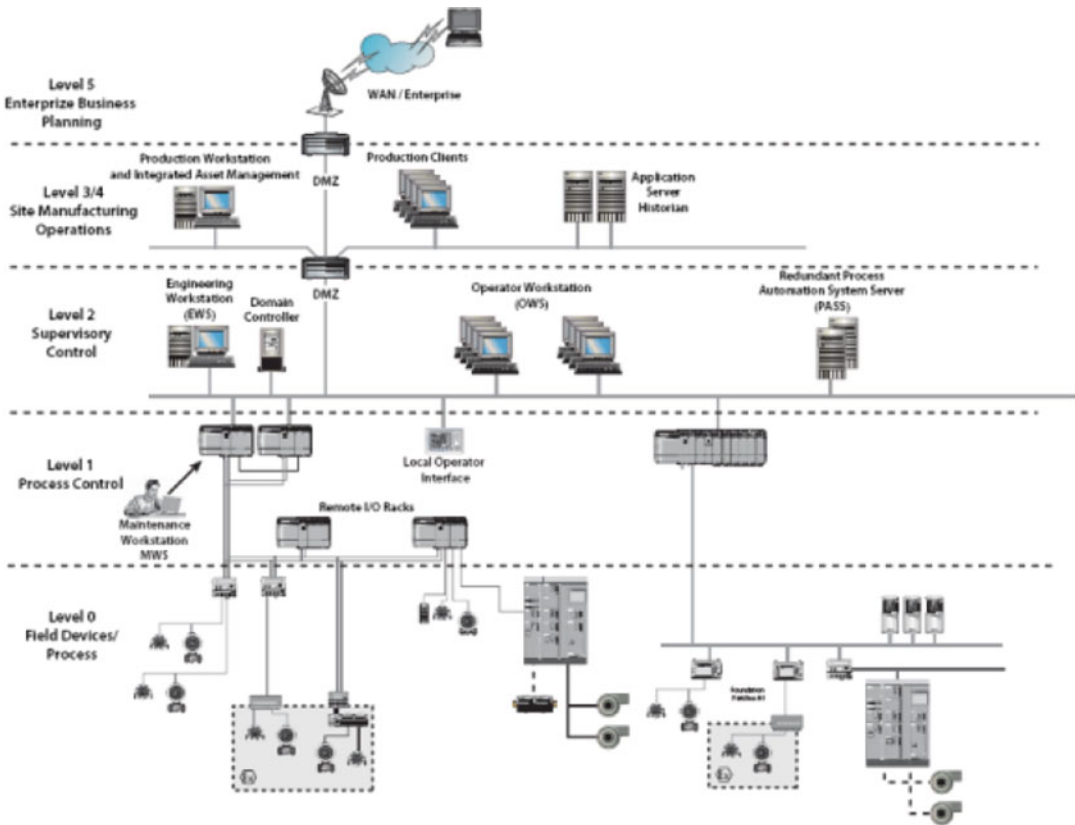
**Control Hierarchy of Large Processing Plants: An Overview, Fig. 3** Information network in a modern process plant

supervision (▶ Multiscale Multivariate Statistical Process Control), data reconciliation, inferences and estimation of unmeasured quantities, fault detection and diagnosis, or performance controller monitoring (▶ Controller Performance Monitoring) (CPM).

The information flow becomes more complex when we move up the basic control layer, looking more like a web than a pyramid when we enter the world of what can be called generally as asset (plant and equipment) management: a collection of different activities oriented to sustain performance and economic return, considering their entire cycle of life and, in particular, aspects such as maintenance, efficiency, or production organization. Above the supervisory layer, one can usually distinguish at least two levels denoted generically as manufacturing execution systems (MES) and enterprise resource planning (ERP) (Scholten 2009) as can be seen in Fig. 4.

MES are information systems that support the functions that a production department must perform in order to prepare and to manage work instructions, schedule production activities, monitor the correct execution of the production process, gather and analyze information about the production process, and optimize procedures. Notice that regarding the control of process units, up to this level no fundamental differences appear between continuous and batch processes. But at the MES level, which corresponds to RTO of Fig. 1, many process units may be involved, and the tools and problems are different, the main task in batch production being the optimal scheduling of those process units (▶ Scheduling of Batch Plants; Mendez et al. 2006).

MES are part of a larger class of systems called manufacturing operation management (MOM) that cover not only the management of production operations but also other functions

Control Hierarchy of Large Processing Plants: An Overview, Fig. 4 Software/hardware view

Software / Hardware
Enterprise view
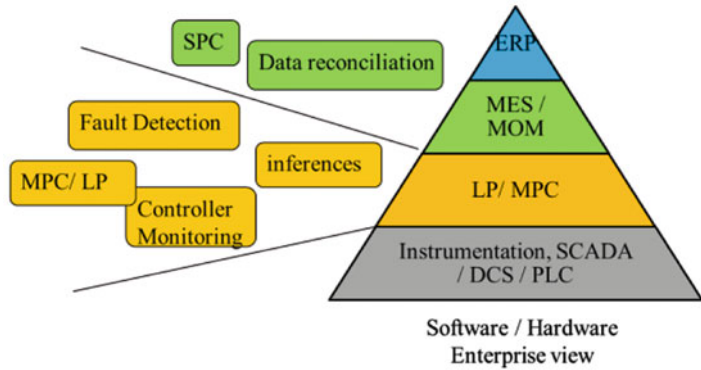
such as maintenance, quality, laboratory information systems, or warehouse management. One of their main tasks is to generate elaborate information, quite often in the form of key performance indicators (KPIs), with the purpose of facilitating the implementation of corrective actions.

ERP systems represent the top of the pyramid, corresponding to the enterprise business planning activities that allows assigning global targets to production scheduling. For many years, it has been considered to be out of the scope of the field of control, but nowadays, more and more, supply chain management is viewed and addressed as a control and optimization problem in research.
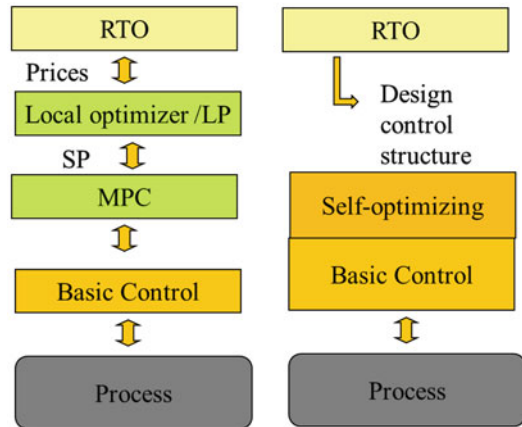
Control Hierarchy of Large Processing Plants: An Overview, Fig. 5 Two possible implementations of RTO

Future Control and Optimization at Plant Scale

Going back to Fig. 1, the variety of control and optimization problems increases as we move up in the control hierarchy, entering the field of dynamic process operations and considering not only individual process units but also larger sets of equipment or whole plants. Examples at the RTO (or MES) level are optimal management of shared resources or utilities, production bottleneck avoidance, optimal energy use or maximum efficiency, smooth transitions against production changes, etc.

Above, we have mentioned RTO as the most common approach for plant-wide optimization. Normally, RTO systems perform the optimization of an economic cost function using a nonlinear

process model in steady state and the corresponding operational constraints to generate targets for the control systems on the lower layers. The implementation of RTO provides consistent benefits by looking at the optimal operation problem from a plant-wide perspective. Nevertheless, in practice, when MPCs with local optimizers are operating the process units, many coordination problems appear between these layers, due to differences in models and targets, so that driving the operation of these process units in a coherent way with the global economic targets is an additional challenge.

A different perspective is taken by the so-called self-optimizing control (Fig. 5 right, Skogestad 2000) that, instead of implementing the RTO solution online, uses it to design a control structure that assures a near optimum
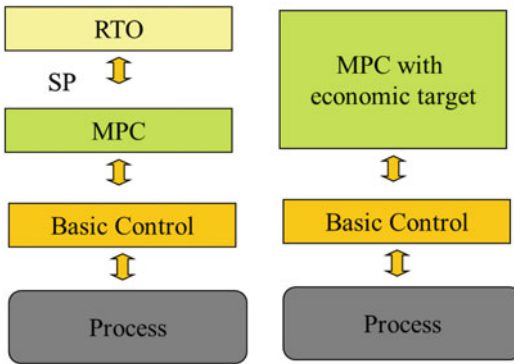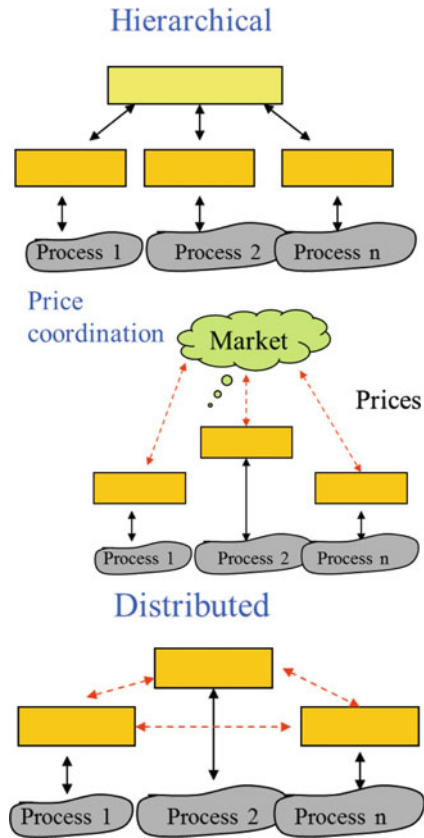
Control Hierarchy of Large Processing Plants: An Overview, Fig. 6 Direct dynamic optimization

operation if some specially chosen variables are maintained closed to their targets.

As in any model-based approach, the problem of how to implement or modify the theoretical optimum computed by RTO so that the optimum computed with the model and the real optimum of the process coincide in spite of model errors, disturbances, etc., emerges. A common choice to deal with this problem is to update periodically the model using parameter estimation methods or data reconciliation with plant data in steady state. Also, uncertainty can be explicitly taken into account by considering different scenarios and optimizing the worst case, but this is conservative and does not take advantage of the plant measurements. Along this line, there are proposals of other solutions such as modifier-adaptation methods that use a fixed model and process measurements to modify the optimization problem so that the final result corresponds to the process optimum (Marchetti et al. 2009) or the use of stochastic optimization where several scenarios are taken into account and future decisions are used as recourse variables (Lucia et al. 2013).

RTO is formulated in steady state, but in practice, most of the time the plants are in transients, and there are many problems, such as start-up optimization, that require a dynamic formulation. A natural evolution in this direction is to combine nonlinear MPC with economic optimization so that the target of the NMPC is not set point following but direct economic optimization as in the right-hand side of Fig. 6: ▸ Economic Model

Control Hierarchy of Large Processing Plants: An Overview, Fig. 7 Hierarchical, price coordination, and distributed approaches

Predictive Control and ▸ Model-Based Performance Optimizing Control (Engell 2007).

The type of problems that can be formulated within this framework is very wide, as are the possible fields of application. Processes with distributed parameter structure or mixtures of real and on/off variables, batch and continuous units, statistical distribution of particle sizes or properties, etc., give rise to special type of NMPC problems (see, e.g., Lunze and Lamnabhi-Lagarrigue 2009), but a common characteristic of all of them is the fact that they are computational intensive and should be solved taking into account the different forms of uncertainty always present.

Control and optimization are nowadays inseparable essential parts of any advanced approach to dynamic process operation. Progress in the field and spreading of the

industrial applications are possible thanks to the advances in optimization methods and tools and computing power available on the plant level, but implementation is still a challenge from many points of view, not only technical. Few suppliers offer commercial products, and finding optimal operation policies for a whole factory is a complex task that requires taking into consideration many aspects and elaborate information not available directly as process measurements. Solving large NMPC problems in real time may require breaking the associated optimization problem in subproblems that can be solved in parallel. This leads to several local controllers/optimizers, each one solving one subproblem involving variables of a part of the process and linked by some type of coordination. This offers a new point of view of the control hierarchy. Typically, three types of architectures are mentioned for dealing with this problem, represented in Fig. 7: In the hierarchical approach, coordination between local controllers is made by an upper layer that deals with the interactions, assigning targets to them. In price coordination, the coordination task is performed by a market-like mechanism that assigns different prices to the cost functions of every local controller/optimizer. Finally, in the distributed approach, the local controllers coordinate their actions by interchanging information about its decisions or states with neighbors (Scattolini 2009).

Summary and Future Research

Process control is a key element in the operation of process plants. At the lowest layer, it can be considered a mature, well-proven technology, even if many problems such as control structure selection and controller tuning in reality are often not solved well. The range of problems under consideration is continuously expanding to the upper layers of the hierarchy, merging control with process operation and optimization, creating new challenges that range from modeling and estimation to efficient large-scale optimization

and robustness against uncertainty, and leading to new challenges and problems for research and possibly large improvements of plant operations.

Cross-References

▶ Controller Performance Monitoring
▶ Economic Model Predictive Control
▶ Industrial MPC of Continuous Processes
▶ Model-Based Performance Optimizing Control
▶ Multiscale Multivariate Statistical Process Control
▶ Programmable Logic Controllers
▶ Real-Time Optimization of Industrial Processes
▶ Scheduling of Batch Plants

Bibliography

Camacho EF, Bordóns C (2004) Model predictive control. Springer, London, pp 1–405. ISBN: 1-85233-694-3

de Prada C, Gutierrez G (2012) Present and future trends in process control. Ingeniería Química 44(505):38–42. Special edition ACHEMA, ISSN:0210–2064

Engell S (2007) Feedback control for optimal process operation. J Process Control 17:203–219

Engell S, Harjunkoski I (2012) Optimal operation: scheduling, advanced control and their integration. Comput Chem Eng 47:121–133

Lucia S, Finkler T, Engell S (2013) Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. J Process Control 23:1306–1319

Lunze J, Lamnabhi-Lagarrigue F (2009) HYCON handbook of hybrid systems control. Theory, tools, applications. Cambridge University Press, Boca Raton. ISBN:978-0-521-76505-3

Marchetti A, Chachuat B, Bonvin D (2009) Modifier-adaptation methodology for real-time optimization. Ind Eng Chem Res 48(13):6022–6033

Mendez CA, Cerdá J, Grossmann I, Harjunkoski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput Chem Eng 30:913–946

Scattolini R (2009) Architectures for distributed and hierarchical model predictive control – a review. J Process Control 19:723–731

Scholten B (2009) MES guide for executives: why and how to select, implement, and maintain a manufacturing execution system. ISA, Research Triangle Park. ISBN:978-1-936007-03-5

C

Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. J Process Control 10:487–507

Control of Biotechnological Processes

Rudibert King
Technische Universität Berlin, Berlin, Germany

Abstract

Closed-loop control can significantly improve the performance of bioprocesses, e.g., by an increase of the production rate of a target molecule or by guaranteeing reproducibility of the production with low variability. In contrast to the control of chemical reaction systems, the biological reactions take place inside cells which constitute highly regulated, i.e., internally controlled systems by themselves. As a result, through evolution, the same cell can and will mimic a system of first order in some situations and a high-dimensional, highly nonlinear system in others. A complete mathematical description of the possible behaviors of the cell is still beyond reach and would be far too complicated as a basis for model-based process control. This makes supervision, control, and optimization of biosystems very demanding.

Keywords

Bioprocess control; Control of uncertain systems; Optimal control; Parameter identification; State estimation; Structure identification; Structured models

Introduction

Biotechnology offers solutions to a broad spectrum of challenges faced today, e.g., for health care, remediation of environmental pollution, new sources for energy supplies, sustainable food production, and the supply of bulk chemicals. To explain the needs for control of bioprocesses, especially for the production of high-value and/or large-volume compounds, it is instructive to have a look on the development of a new process. If a potential strain is found or genetically engineered, the biologist will determine favorable environmental factors for the growth of and the production of the target product by the cells. These factors typically comprise the levels of temperature, pH, dissolved oxygen, etc. Moreover, concentration regions for the nutrients, precursors, and so-called trace elements are specified. Whereas for the former variables often "optimal" setpoints are provided which, at least in smaller scale reactors, can be easily maintained by independent classically designed controllers, information about the best nutrient supply is incomplete from a control engineering point of view. It is this dynamic nutrient supply which is most often not revealed in the biological laboratory and which, however, offers substantial room for production improvements by control.

Irrespective whether bacteria, yeasts, fungi, or animal cells are used for production, these cells will consist of thousands of different compounds which react with each other in hundreds or more reactions. All reactions are tightly regulated on a molecular and genetic basis; see ▶ Deterministic Description of Biochemical Networks. For so-called unlimited growth conditions, all cellular compartments will be built up with the same specific growth rate, meaning that the cellular composition will not change over time. In a mathematical model describing growth and production, only one state variable will be needed to describe the biotic phase. This will give rise to unstructured models; see below. Whenever a cell enters a limitation, which is often needed for production, the cell will start to reorganize its internal reaction pathways. Model-based approaches of supervision and control based on unstructured models are now bound to fail. More biotic state variables are needed. However, it is not clear which and how many. As a result, modeling of limiting behaviors is challenging and crucial for the control of biotechnological processes. It requires a large amount of process-specific information. Moreover, model-based estimates of

the state of the cell and of the environment are a key factor as online measurements of the internal processes in the cell and of the nutrient concentrations are usually impossible. Finally, as models used for process control have to be limited in size and thus only give an approximative description, robustness of the methods has to be addressed.

Mathematical Models

For the production of biotechnical goods, many up- and downstream unit operations are involved besides the biological reactions. As these pose no typical bio-related challenges, we will concentrate here on the cultivation of the organisms only. This is mostly performed in aqueous solutions in special bioreactors through which air is sparged for a supply with oxygen. In some cases, other gases are supplied as well; see Fig. 1. Disregarding wastewater treatment plants, most cultivations are still performed in a fed-batch mode, meaning that a small amount of cells and part of the nutrients are put into the reactor initially. Then more nutrients and correcting fluids, e.g.,

for pH or antifoam control, are added with variable rates leading to an unsteady behavior. The system to be modeled consists of the gaseous, the liquid, and the biotic phase inside the reactor. For the former ones, balance equations can be formulated readily. The biotic phase can be modeled in a structured or unstructured way. Moreover, as not all cells behave similarly, this may give rise to a segregated model formulation which is omitted here for brevity.

Unstructured Models
If the biotic phase is represented by just one state variable, m_X, a typical example of a simple unstructured model of the liquid phase would be
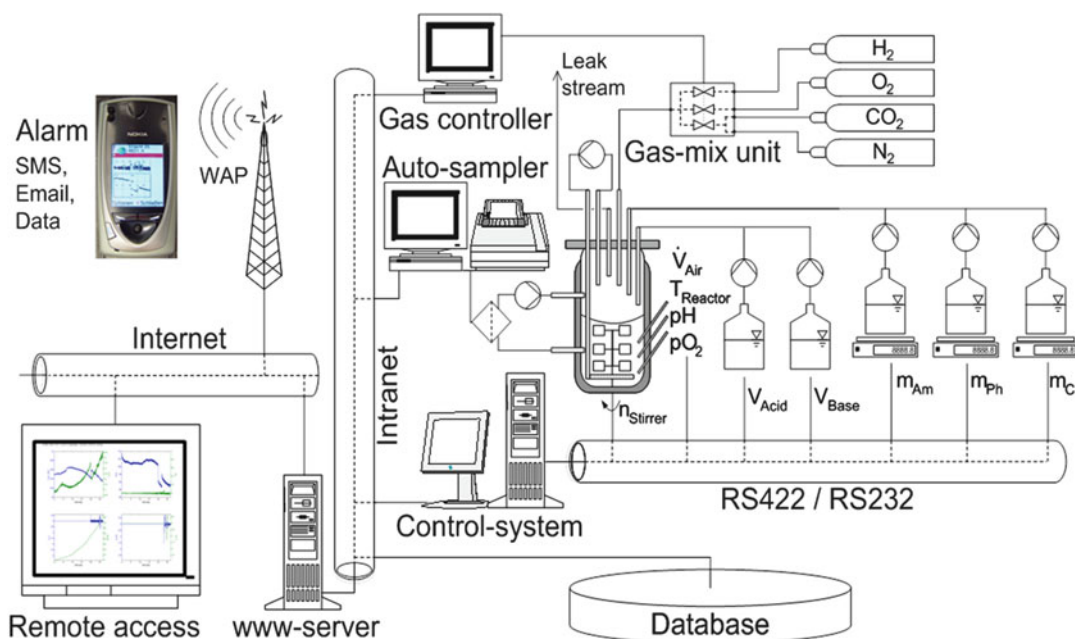
$$\dot{m}_X = \mu_X m_X$$
$$\dot{m}_P = \mu_P m_X$$
$$\dot{m}_S = -a_1 \mu_X m_X - a_2 \mu_P m_X + c_{S,feed} u$$
$$\dot{m}_O = a_3(a_4 - c_O) - a_5 \mu_X m_X - a_6 \mu_P m_X$$
$$\dot{V} = u$$

Control of Biotechnological Processes, Fig. 1 Modern laboratory reactor platform for control-oriented process development

Control of Biotechnological Processes, Table 1 Multiplicative rates depending on several concentrations $c_1, \ldots, c_k$ with possible kinetic terms

$\mu_i = a_{imax}\mu_{i1}(c_1) \cdot \mu_{i2}(c_2) \cdot \ldots \cdot \mu_{ik}(c_k)$		
μ_{ij}		
$\dfrac{c_j}{c_j + a_{ij}}$	$\dfrac{a_{ij}}{c_j + a_{ij}}$	$\dfrac{c_j}{c_j^2 + a_{ij}}$
$\dfrac{c_j}{c_j + a_{ij}}e^{-c_j/a_{ij+1}}$	$\dfrac{c_j}{a_{ij}c_j^2 + c_j + a_{ij+1}}$	$\ldots$

with the masses m_i with $i = X, P, S, O$ for cells, product, substrate or nutrient, and dissolved oxygen, respectively. The volume is given by V, and the specific growth and production rates μ_X and μ_P depend on concentrations $c_i = m_i/V$, e.g., of the substrate S or oxygen O according to formal kinetics, e.g.,

$$\mu_X = \frac{a_7 c_S c_O}{(c_S + a_8)(c_O + a_9)}$$
$$\mu_P = \frac{a_{10} c_S}{a_{11} c_S^2 + c_S + a_{12}}$$

The nutrient supply can be changed by the feed rate $u(t)$ as a control input, with inflow concentration $c_{S,feed}$. Very often, just one feed stream is considered in unstructured models. As all parameters a_i have to be identified from noisy and infrequently sampled data, a low-dimensional nonlinear uncertain model results. All steps prior to the cultivation in which, e.g., from frozen cells, enough cells are produced to start the fermentation add to the uncertainty. Whereas the balance equations follow from first principles-based modeling, the structure of the kinetics μ_X and μ_P is unknown, i.e., empirical relations are exploited. Many different kinetic expressions can be used here; see Bastin and Dochain (1990) or a small selection shown in Table 1.

It has to be pointed out that, most often, neither c_X, c_P, nor c_S are measured online. As the measurement of c_O might be unreliable, the exhaust gas concentration of the gaseous phase is the main online measurement which can be used by employing an additional balance equation for the gaseous phase. Infrequent at-line measurements, though, are sometimes available for X, P, S, especially at the lab-scale during process development.

Structured Models

In structured models, the changing composition and reaction pathways of the cell is accounted for. As detailed information about the cell's complete metabolism including all regulations is missing for the majority if not all cells exploited in bioprocesses, an approximative description is used. Examples are models in which a part of the real metabolism is described on a mechanistic level, whereas the rest is lumped together into one or very few states (Goudar et al. 2006), cybernetic models (Varner and Ramkrishna 1998), or compartment models (King 1997). As an example, all compartment models can be written down as

$$\underline{\dot{m}} = \mathbf{A}\underline{\mu}(\underline{c}) + \underline{f}_{in}(\underline{u}) + \underline{f}_{out}(\underline{u})$$
$$\dot{V} = \sum_i u_i$$

with vectors of streams into and out of the reaction mixture, $\underline{f}_{in}$ and $\underline{f}_{out}$, which depend on control inputs $\underline{u}$; a matrix of (stoichiometric) parameters, $\mathbf{A}$; a vector of reaction rates $\underline{\mu} = \underline{\mu}(\underline{c})$; and, finally, a vector $\underline{m}$ comprising substrates, products, and more than one biotic state. These biotic states can be motivated, for example, by physiological arguments, describing the total amounts of macromolecules in the cell, such as the main building blocks DNA, RNA, and proteins. In very simple compartment models, the cell is only divided up into what is called active and inactive biomass. Again, all coefficients in $\mathbf{A}$ and the structure and the coefficients of all entries in $\underline{\mu}(\underline{c})$ (see Table 1) are unknown and have to be identified based on experimental data. Issues of structural and practical identifiability are of major concern. For models of system biology (see ▶ Deterministic Description of Biochemical Networks), algebraic equations are

added that describe the dependencies between individual fluxes. Then at least part of **A** is known.

Describing the biotic phase with a higher degree of granularity does not change the measurement situation in the laboratory or in the production scale, i.e., still only very few online measurements will be available for control.

Identification

Even if the growth medium initially "only" consists of some 10–20 different, chemically well-defined substances, from which only few are described in the model, this situation will change over the cultivation time as the organisms release further compounds from which only few may be known. If, for economic reasons, complex raw materials are used, even the initial composition is unknown. Hence, measuring the concentrations of some of the compounds of the large set of substances as a basis for modeling is not trivial. For structured models, intracellular substances have to be determined additionally. These are embedded in an even larger matrix of compounds making chemical analysis more difficult. Therefore, the basis for parameter and structure identification is uncertain.

As the expensive experiments and chemical analysis tasks are very time consuming, sometimes lasting up to several weeks, methods of optimal experimental design should always be considered in biotechnology; see ▶ Experiment Design and Identification for Control.

The models to be built up should possess some predictive capability for a limited range of environmental conditions. This rules out unstructured models for many practical situations. However, for process control, the models should still be of manageable complexity. Medium-sized structured models seem to be well suited for such a situation. The choice of biotic states in $\underline{m}$ and possible structures for the reaction rates μ_i, however, is hardly supported by biological or chemical evidence. As a result, a combined structure and parameter identification problem has to be solved. The choices of possible terms μ_{ij} in all μ_i give rise to a problem that exhibits

a combinatorial explosion. Although approaches exist to support this modeling step (see Herold and King 2013 or Mangold et al. 2005) finally, the modeler will have to settle with a compromise with respect to the accuracy of the model found versus the number of fully identified model candidates. As a result, all control methods applied should be robust in some sense.

Soft Sensors

Despite many advantages in the development of online measurements (see Mandenius and Titchener-Hooker 2013) systems for supervision and control of biotechnical processes often include model-based estimations schemes, such as extended Kalman filters (EKF); see ▶ Kalman Filters. Concentration estimates are needed for unmeasured substances and for quantities which depend on these concentrations like the growth rate of the cells. In real applications, formulations have to be used which account for delays in laboratory analysis of up to several hours and for situations in which results from the laboratory will not be available in the same sequence as the samples were taken. An example from a real cultivation is shown in Fig. 2. Here, the at-line measurement of the biomass concentration, $c_X = m_X / V$, is the only measurement available. The result of a single measurement is obtained about 30 min after sampling. For reference, unaccessible state variables, which were analyzed later, are shown as well along with the online estimates. The scatter of the data, especially of DNA and RNA, gives a qualitative impression of the measurement accuracy in biotechnology.

Control

Beside the relatively simple control of physical parameters, such as temperature, pH, dissolved oxygen, or carbon dioxide concentration, only few biotic variables are typically controlled with respect to a setpoint. The most prominent example is the growth rate of the biomass with the goal to reach a high cell concentration
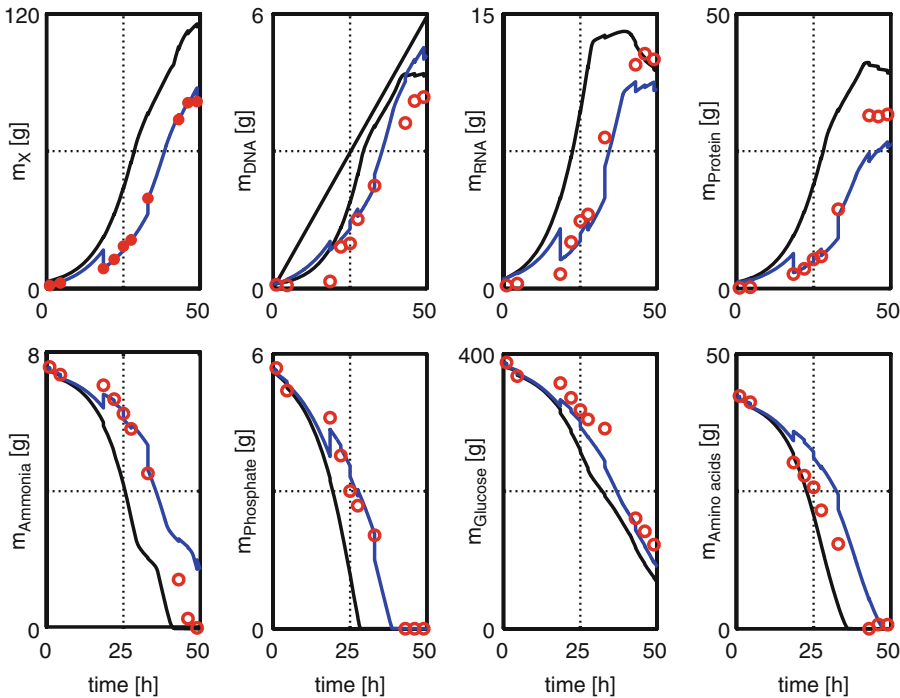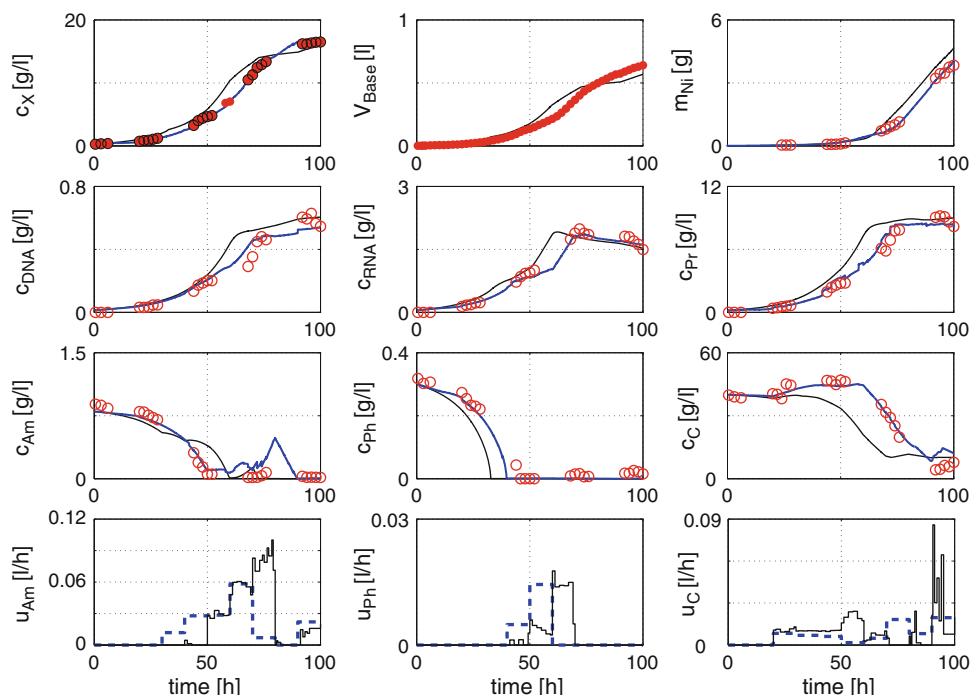
Control of Biotechnological Processes, Fig. 2 Estimation of states of a structured model with an EKF with an unexpected growth delay initially. At-line measurement m_X (*red filled circles*), initially predicted evolution of states (*black*), online estimated evolution (*blue*), off-line data analyzed after the experiment (*open red circles*) (Data obtained by T. Heine)

in the reactor as fast as possible. This is the predominant goal when the cells are the primary target as in baker's yeast cultivations or when the expression of the desired product is growth associated. For other non-growth-associated products, a high cell mass is desirable as well, as production is proportional to the amount of cells. If the nutrient supply is maintained above a certain level, unlimited growth behavior results, allowing the use of unstructured models for model-based control. An excess of nutrients has to be avoided, though, as some organisms, like baker's yeast, will initiate an overflow metabolism, with products which may be inhibitory in later stages of the cultivation. For some products, such as the antibiotic penicillin, the organism has to grow slowly to obtain a high production rate. For these so-called secondary metabolites, low but not vanishing concentrations for some limiting substrates are needed. If setpoints are given for these

concentrations instead, this can pose a rather challenging control problem. As the organisms try to grow exponentially, the controller must be able to increase the feed exponentially as well. The difficulty mainly arises from the inaccurate and infrequent measurements that the soft sensors/controller has to work with and from the danger that an intermediate shortage or oversupply with nutrients may switch the metabolism to an undesired state of low productivity.

For control of biotechnical processes, many methods explained in this encyclopedia including feedforward, feedback, model-based, optimal, adaptive, fuzzy, neural nets, etc., can be and have been used (cf. Dochain 2008; Gnoth et al. 2008; Rani and Rao 1999). As in other areas of application, (robust) model-predictive control schemes (MPC) (see ▶ Industrial MPC of Continuous Processes) are applied with great success in biotechnology.

Control of Biotechnological Processes, Fig. 3 MPC control and state estimation of a cultivation with *S. tendae*. At-line measurement m_X (*red filled circles*), initially predicted evolution of states (*black*), online estimated evolution (*blue*), off-line data analyzed after the experiment (*open red circles*). Off-line optimal feeding profiles u_i (*blue broken line*), MPC-calculated feeds (*black, solid*) (Data obtained by T. Heine)

For the antibiotic production shown in Fig. 3, optimal feeding profiles u_i for ammonia (AM), phosphate (PH), and glucose (C) were calculated before the experiment was performed in a trajectory optimization such that the final mass of the desired antibiotic nikkomycin (Ni) was maximized. This resulted in the blue broken lines for the feeds u_i. However, due to disturbances and model inaccuracies, an MPC scheme had to significantly change the feeding profiles, to actually obtain this high amount of nikkomycin; see the feeding profiles given in black solid lines. This example shows that, especially in biotechnology, off-line trajectory planning has to be complemented by closed-loop concepts.

On the other hand, the experimental data given in Fig. 2 shows that significant disturbances, such as an unexpected initial growth delay, may occur in real systems as well. For this reason, the classical receding horizon MPC with an off-line determined optimal reference trajectory will not always be the best solution, and an online optimization over the whole horizon has a larger potential (cf. Kawohl et al. 2007).

Summary and Future Directions

Advanced process control including soft sensors can significantly improve biotechnical processes. Using these techniques promotes quality and reproducibility of processes (Junker and Wang 2006). These methods should, however, not only be exploited in the production scale. For new pharmaceutical products, the time to market is the decisive factor. Methods of (model-based) monitoring and control can help here to speed up process development. Since a few years, a clear trend can be seen in biotechnology to miniaturize and parallelize process development using multi-fermenter systems and robotic tech-

nologies. This trend gives rise to new challenges for modeling on the basis of huge data sets and for control in very small scales. At the same time, it is expected that a continued increase of information from bioinformatic tools will be available which has to be utilized for process control as well. Going to large-scale cultivations adds further spatial dimensions to the problem. Now, the assumption of a well-stirred, ideally mixed reactor does not longer hold. Substrate concentrations will be space dependent. Cells will experience changing good and bad nutrient environments frequently. Thus, mass transfer has to be accounted for, leading to partial differential equations as models for the process.

Cross-References

Bibliography

Bastin G, Dochain D (2008) On-line estimation and adaptive control of bioreactors. Elsevier, Amsterdam

Dochain D (2008) Bioprocess control. ISTE, London

Gnoth S, Jentzsch M, Simutis R, Lübbert A (2008) Control of cultivation processes for recombinant protein production: a review. Bioprocess Biosyst Eng 31:21–39

Goudar C, Biener R, Zhang C, Michaels J, Piret J, Konstantinov K (2006) Towards industrial application of quasi real-time metabolic flux analysis for mammalian cell culture. Adv Biochem Eng Biotechnol 101:99–118

Herold S, King R (2013) Automatic identification of structured process models based on biological phenomena detected in (fed-)batch experiments. Bioprocess Biosyst Eng. doi:10.1007/s00449-013-1100-6

Junker BH, Wang HY (2006) Bioprocess monitoring and computer control: key roots of the current PAT initiative. Biotechnol Bioeng 95:226–261

Kawohl M, Heine T, King R (2007) Model-based estimation and optimal control of fed-batch fermentation processes for the production of antibiotics. Chem Eng Process 11:1223–1241

King R (1997) A structured mathematical model for a class of organisms: part 1–2. J Biotechnol 52:219–244

Mandenius CF, Titchener-Hooker NJ (ed) (2013) Measurement, monitoring, modelling and control of bioprocesses. Advances in biochemical engineering biotechnology, vol 132. Springer, Heidelberg

Mangold M, Angeles-Palacios O, Ginkel M, Waschler R, Kinele A, Gilles ED (2005) Computer aided modeling of chemical and biological systems – methods, tools, and applications. Ind Eng Chem Res 44:2579–2591

Rani KY, Rao VSR (1999) Control of fermenters. Bioprocess Eng 21:77–88 31:21–39

Varner J, Ramkrishna D (1998) Application of cybernetic models to metabolic engineering: investigation of storage pathways. Biotech Bioeng 58:282–291; 31:21–39

Control of Fluids and Fluid-Structure Interactions

Jean-Pierre Raymond
Institut de Mathématiques, Université Paul Sabatier Toulouse III & CNRS, Toulouse Cedex, France

Abstract

We introduce control and stabilization issues for fluid flows along with known results in the field. Some models coupling fluid flow equations and equations for rigid or elastic bodies are presented, together with a few controllability and stabilization results.

Keywords

Control; Fluid flows; Fluid-structure systems; Stabilization

Some Fluid Models

We consider a fluid flow occupying a bounded domain $\Omega_F \subset \mathbb{R}^N$, with $N = 2$ or $N = 3$, at the initial time $t = 0$, and a domain $\Omega_F(t)$

at time $t > 0$. Let us denote by $\rho(x, t) \in \mathbb{R}^+$ the density of the fluid at time t at the point $x \in \Omega_F(t)$ and by $u(x, t) \in \mathbb{R}^N$ its velocity. The fluid flow equations are derived by writing the mass conservation

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho u) = 0 \quad \text{in } \Omega_F(t), \quad \text{for } t > 0, \ (1)$$

and the balance of momentum

$$\rho\left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u\right) = \operatorname{div}\sigma + \rho\, f \qquad (2)$$
$$\text{in } \Omega_F(t), \quad \text{for } t > 0$$

where σ is the so-called constraint tensor and f represents a volumic force. For an isothermal fluid, there is no need to complete the system by the balance of energy. The physical nature of the fluid flow is taken into account in the choice of the constraint tensor σ. When the volume is preserved by the fluid flow transport, the fluid is called incompressible. The incompressibility condition reads as $\operatorname{div} u = 0$ in $\Omega_F(t)$. The incompressible Navier-Stokes equations are the classical model to describe the evolution of isothermal incompressible and Newtonian fluid flows. When in addition the density of the fluid is assumed to be constant, $\rho(x, t) = \rho_0$, the equations reduce to

$$\operatorname{div} u = 0,$$
$$\rho_0\left(\frac{\partial u}{\partial t} + (u \cdot \nabla)u\right) = \nu\Delta u - \nabla p + \rho_0\, f$$
$$\text{in } \Omega_F(t), \quad t > 0, \qquad (3)$$

which are obtained by setting

$$\sigma = \nu\left(\nabla u + (\nabla u)^T\right) + \left(\mu - \frac{2\nu}{3}\right)\operatorname{div} u\, I - pI, \qquad (4)$$

in Eq. (2). When $\operatorname{div} u = 0$, the expression of σ simplifies. The coefficients $\nu > 0$ and $\mu > 0$ are the viscosity coefficients of the fluid, and $p(x, t)$ its pressure at the point $x \in \Omega_F(t)$ and at time $t > 0$.

This model has to be completed with boundary conditions on $\partial\Omega_F(t)$ and an initial condition at time $t = 0$.

The incompressible Euler equations with constant density are obtained by setting $\nu = 0$ in the above system.

The compressible Navier-Stokes system is obtained by coupling the equation of conservation of mass Eq. (1) with the balance of momentum Eq. (2), where the tensor σ is defined by Eq. (4), and by completing the system with a constitutive law for the pressure.

Control Issues

There are unstable steady states of the Navier-Stokes equations which give rise to interesting control problems (e.g., to maximize the ratio "lift over drag"), but which cannot be observed in real life because of their unstable nature. In such situations, we would like to maintain the physical model close to an unstable steady state by the action of a control expressed in feedback form, that is, as a function either depending on an estimation of the velocity or depending on the velocity itself. The estimation of the velocity of the fluid may be recovered by using some real-time measurements. In that case, we speak of a feedback stabilization problem with partial information. Otherwise, when the control is expressed in terms of the velocity itself, we speak of a feedback stabilization problem with full information.

Another interesting issue is to maintain a fluid flow (described by the Navier-Stokes equations) in the neighborhood of a nominal trajectory (not necessarily a steady state) in the presence of perturbations. This is a much more complicated issue which is not yet solved.

In the case of a perturbation in the initial condition of the system (the initial condition at time $t = 0$ is different from the nominal velocity held at time $t = 0$), the exact controllability to the nominal trajectory consists in looking for controls driving the system in finite time to the desired trajectory.

Thus, control issues for fluid flows are those encountered in other fields. However there are

specific difficulties which make the corresponding problems challenging. When we deal with the incompressible Navier-Stokes system, the pressure plays the role of a Lagrange multiplier associated with the incompressibility condition. Thus, we have to deal with an infinite-dimensional nonlinear differential algebraic system. In the case of a Dirichlet boundary control, the elimination of the pressure, by using the so-called Leray or Helmholtz projector, leads to an unusual form of the corresponding control operator; see Raymond (2006). In the case of an internal control, the estimation of the pressure to prove observability inequalities is also quite tricky; see Fernandez-Cara et al. (2004). From the numerical viewpoint, the approximation of feedback control laws leads to very large-size problems, and new strategies have to be found for tackling these issues.

Moreover, the issues that we have described for the incompressible Navier-Stokes equations may be studied for other models like the compressible Navier-Stokes equations, the Euler equations (describing nonviscous fluid flows) both for compressible and incompressible models, or even more complicated models.

Feedback Stabilization of Fluid Flows

Let us now describe what are the known results for the incompressible Navier-Stokes equations in 2D or 3D bounded domains, with a control acting locally in a Dirichlet boundary condition. Let us consider a given steady state (u_s, p_s) satisfying the equation

$$-\nu \Delta u_s + (u_s \cdot \nabla) u_s + \nabla p_s = f_s,$$
$$\text{and} \quad \text{div } u_s = 0 \quad \text{in} \quad \Omega_F,$$

with some boundary conditions which may be of Dirichlet type or of mixed type (Dirichlet-Neumann-Navier type). For simplicity, we only deal with the case of Dirichlet boundary conditions

$$u_s = g_s \quad \text{on} \quad \partial \Omega_F,$$

where g_s and f_s are time-independent functions. In the case $\Omega_F(t) = \Omega_F$, not depending on t, the corresponding instationary model is

$$
\begin{aligned}
&\frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u + \nabla p = f_s \\
&\text{and} \quad \text{div } u = 0 \text{ in } \Omega_F \times (0, \infty), \\
&u = g_s + \sum_{i=1}^{N_c} f_i(t) g_i, \ \partial \Omega_F \times (0, \infty) \\
&u(0) = u_0 \quad \text{on } \Omega_F.
\end{aligned}
\tag{5}
$$

In this model, we assume that $u_0 \neq u_s$, g_i are given functions with localized supports in $\partial \Omega_F$ and $f(t) = (f_1(t), \ldots, f_{N_c}(t))$ is a finite-dimensional control. Due to the incompressibility condition, the functions g_i have to satisfy

$$\int_{\partial \Omega_F} g_i \cdot n = 0,$$

where n is the unit normal to $\partial \Omega_F$, outward Ω_F.

The stabilization problem, with a prescribed decay rate $-\alpha < 0$, consists in looking for a control f in feedback form, that is, of the form

$$f(t) = K(u(t) - u_s), \tag{6}$$

such that the solution to the Navier-Stokes system Eq. (5), with f defined by Eq. (6), obeys

$$\left\| e^{\alpha t} (u(t) - u_s) \right\|_z \le \varphi \left(\|u_0 - u_s\|_z \right),$$

for some norm Z, provided $\|u_0 - u_s\|_z$ is small enough and where φ is a nondecreasing function. The mapping K, called the feedback gain, may be chosen linear.

The usual procedure to solve this stabilization problem consists in writing the system satisfied by $u - u_s$, in linearizing this system, and in looking for a feedback control stabilizing this linearized model. The issue is first to study the stabilizability of the linearized model and, when it is stabilizable, to find a stabilizing feedback gain. Among the feedback gains that stabilize the linearized model, we have to find one able to stabilize, at least locally, the nonlinear system too.

The linearized controlled system associated with Eq. (5) is

$$\frac{\partial v}{\partial t} - \nu \Delta v + (u_s \cdot \nabla)v + (v \cdot \nabla)u_s + \nabla q = 0$$
and   div $v = 0$ in $\Omega_F \times (0, \infty)$,
$v = \sum_{i=1}^{N_c} f_i(t)g_i$ on $\partial\Omega_F \times (0, \infty)$,
$v(0) = v_0$ on Ω_F.
(7)

The easiest way for proving the stabilizability of the controlled system Eq. (7) is to verify the Hautus criterion. It consists in proving the following unique continuation result. If $(\phi_j, \psi_j, \lambda_j)$ is the solution to the eigenvalue problem

$$\lambda_j \phi_j - \nu \Delta \phi_j - (u_s \cdot \nabla)\phi_j + (\nabla u_s)^T \phi_j$$
$$+ \nabla \psi_j = 0 \quad \text{and} \quad \text{div} \phi_j = 0 \text{ in } \Omega_F,$$
$$\phi_j = 0 \text{ on } \partial\Omega_F, \quad \text{Re } \lambda_j \geq -\alpha, \quad (8)$$

and if in addition (ϕ_j, ψ_j) satisfies

$$\int_{\partial\Omega_F} g_i \cdot \sigma(\phi_j, \psi_j)n = 0 \quad \text{for all } 1 \leq i \leq N_c,$$

then $(\phi_j, \psi_j) = 0$. By using a unique continuation theorem due to Fabre and Lebeau (1996), we can explicitly determine the functions g_i so that this condition is satisfied; see Raymond and Thevenet (2010). For feedback stabilization results of the Navier-Stokes equations in two or three dimensions, we refer to Fursikov (2004), Raymond (2006), Barbu et al. (2006), Raymond (2007), Badra (2009), and Vazquez and Krstic (2008).

Controllability to Trajectories of Fluid Flows

If $(\tilde{u}(t), \tilde{p}(t))_{0 \leq t < \infty}$ is a solution to the Navier-Stokes system, the controllability problem to the trajectory $(\tilde{u}(t), \tilde{p}(t))_{0 \leq t < \infty}$, in time $T > 0$, may be rewritten as a null controllability problem satisfied by $(v, q) = (u - \tilde{u}, p - \tilde{p})$. The local null controllability in time $T > 0$ follows from the null controllability of the linearized system and from a fixed point argument. The linearized controlled system is

$$\frac{\partial v}{\partial t} - \nu \Delta v + (\tilde{u}(t) \cdot \nabla)v + (v \cdot \nabla)\tilde{u}(t) + \nabla q = 0$$
and   div $v = 0$ in $\Omega_F \times (0, T)$,
$v = m_c f$ on $\partial\Omega_F \times (0, T)$,
$v(0) = v_0 \in L^2(\Omega_F; \mathbb{R}^N)$,   div $v_0 = 0$.
(9)

The nonnegative function m_c is used to localize the boundary control f. The control f is assumed to satisfy

$$\int_{\partial\Omega_F} m_c f \cdot n = 0. \quad (10)$$

As for general linear dynamical systems, the null controllability of the linearized system follows from an observability inequality for the solutions to the following adjoint system

$$-\frac{\partial \phi}{\partial t} - \nu \Delta \phi - (\tilde{u}(t) \cdot \nabla)\phi + (\nabla \tilde{u}(t))^T \phi + \nabla \psi = 0$$
and div $\phi = 0$ in $\Omega_F \times (0, T)$,
$\phi = 0$ on $\partial\Omega_F \times (0, T)$,
$\phi(T) \in L^2(\Omega_F; \mathbb{R}^N)$,   div $\phi(T) = 0$.
(11)

Contrary to the stabilization problem, the null controllability by a control of finite dimension seems to be out of reach and it will be impossible in general. We look for a control $f \in L^2(\partial\Omega_F; \mathbb{R}^N)$, satisfying Eq. (10), driving the solution to system Eq. (9) in time T to zero, that is, such that the solution $v_{v_0, f}$ obeys $v_{v_0, f}(T) = 0$. The linearized system Eq. (9) is null controllable in time $T > 0$ by a boundary control $f \in L^2(\partial\Omega_F; \mathbb{R}^N)$ obeying Eq. (10), if and only if there exists $C > 0$ such that

$$\int_{\Omega_F} |\phi(0)|^2 \, dx \leq C \int_{\partial\Omega_F} m_c |\sigma(\phi, \psi)n|^2 \, dx, \quad (12)$$

for all solution (ϕ, ψ) of Eq. (11). The observability inequality Eq. (12) may be proved by establishing weighted energy estimates called "Carleman-type estimates"; see Fernandez-Cara et al. (2004) and Fursikov and Imanuvilov (1996).

Additional Controllability Results for Other Fluid Flow Models

The null controllability of the 2D incompressible Euler equation has been obtained by J.-M. Coron with the so-called Return Method (Coron 1996). See also Coron (2007) for additional references (in particular, the 3D case has been treated by O. Glass).

Some null controllability results for the one-dimensional compressible Navier-Stokes equations have been obtained in Ervedoza et al. (2012).

Fluid-Structure Models

Fluid-structure models are obtained by coupling an equation describing the evolution of the fluid flow with an equation describing the evolution of the structure. The coupling comes from the balance of momentum and by writing that at the fluid-structure interface, the fluid velocity is equal to the displacement velocity of the structure.

The most important difficulty in studying those models comes from the fact that the domain occupied by the fluid at time t evolves and depends on the displacement of the structure. In addition, when the structure is deformable, its evolution is usually written in Lagrangian coordinates while fluid flows are usually described in Eulerian coordinates.

The structure may be a rigid or a deformable body immersed into the fluid. It may also be a deformable structure located at the boundary of the domain occupied by the fluid.

A Rigid Body Immersed in a Three-Dimensional Incompressible Viscous Fluid

In the case of a 3D rigid body $\Omega_S(t)$ immersed in a fluid flow occupying the domain $\Omega_F(t)$, the motion of the rigid body may be described by the position $h(t) \in \mathbb{R}^3$ of its center of mass and by a matrix of rotation $Q(t) \in \mathbb{R}^{3 \times 3}$.

The domain $\Omega_S(t)$ and the flow X_S associated with the motion of the structure obey

$$
\begin{aligned}
& X_S(y, t) = h(t) + Q(t)Q_0^{-1}(y - h(0)), \\
& \text{for } y \in \Omega_S(0) = \Omega_S, \\
& \Omega_S(t) = X_S(\Omega_S(0), t),
\end{aligned} \tag{13}
$$

and the matrix $Q(t)$ is related to the angular velocity $\omega : (0, T) \mapsto \mathbb{R}^3$, by the differential equation

$$
Q'(t) = \omega(t) \times Q(t), \quad Q(0) = Q_0. \tag{14}
$$

We consider the case when the fluid flow satisfies the incompressible Navier-Stokes system Eq. (3) in the domain $\Omega_F(t)$ corresponding to Fig. 1. Denoting by $J(t) \in \mathbb{R}^{3 \times 3}$ the tensor of inertia at time t, and by m the mass of the rigid body, the equations of the structure are obtained by writing the balance of linear and angular momenta

$$
\begin{aligned}
& mh'' = \int_{\partial \Omega_S(t)} \sigma(u, p)n \, dx, \\
& J\omega' = J\omega \times \omega + \int_{\partial \Omega_S(t)} (x - h) \times \sigma(u, p)n \, dx, \\
& h(0) = h_0, \ h'(0) = h_1, \ \omega(0) = \omega_0,
\end{aligned} \tag{15}
$$

where n is the normal to $\partial \Omega_S(t)$ outward $\Omega_F(t)$. The system Eqs. (3) and (13)–(15) has to be completed with boundary conditions. At the fluid-structure interface, the fluid velocity is equal to the displacement velocity of the rigid solid:

$$
u(x, t) = h'(t) + \omega(t) \times (x - h(t)), \tag{16}
$$

for all $x \in \partial \Omega_S(t), t > 0$. The exterior boundary of the fluid domain is assumed to be fixed
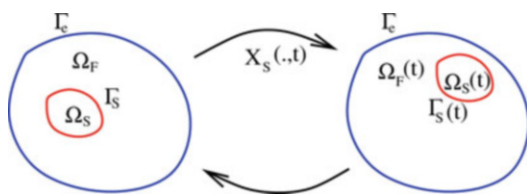
Control of Fluids and Fluid-Structure Interactions, Fig. 1

$\Gamma_e = \partial\Omega_F(t)\backslash\partial\Omega_S(t)$. The boundary condition on $\Gamma_e \times (0, T)$ may be of the form

$$u = m_c\, f \quad \text{on} \ \ \Gamma_e \times (0, \infty), \qquad (17)$$

with $\int_{\Gamma_e} m_c\, f \cdot n = 0$, f is a control, and m_c a localization function.

An Elastic Beam Located at the Boundary of a Two-Dimensional Domain Filled by an Incompressible Viscous Fluid

When the structure is described by an infinite-dimensional model (a partial differential equation or a system of p.d.e.), there are a few existence results for such systems and mainly existence of weak solutions (Chambolle et al. 2005). But for stabilization and control problems of nonlinear systems, we are usually interested in strong solutions. Let us describe a two-dimensional model in which a one-dimensional structure is located on a flat part $\Gamma_S = (0, L) \times \{y_0\}$ of the boundary of the reference configuration of the fluid domain Ω_F. We assume that the structure is a Euler-Bernoulli beam with or without damping. The displacement η of the structure in the direction normal to the boundary Γ_S is described by the partial differential equation

$$\eta_{tt} - b\eta_{xx} - c\eta_{txx} + a\eta_{xxxx} = F, \text{in } \Gamma_S \times (0, \infty),$$
$$\eta = 0 \quad \text{and} \quad \eta_x = 0 \ \text{on} \ \partial\Gamma_S \times (0, \infty),$$
$$\eta(0) = \eta_1^0 \quad \text{and} \quad \eta_t(0) = \eta_2^0 \ \text{in} \ \Gamma_S,$$
$$(18)$$

where η_x, η_{xx}, and η_{xxxx} stand for the first, the second, and the fourth derivative of η with respect to $x \in \Gamma_S$. The other derivatives are defined in a similar way. The coefficients b and c are nonnegative, and $a > 0$. The term $c\eta_{txx}$ is a structural damping term. At time t, the structure occupies the position $\Gamma_S(t) = \{(x, y) \,|\, x \in (0, L), \ y = y_0 + \eta(x, t)\}$. When Ω_F is a two-dimensional model, Γ_S is of dimension one, and $\partial\Gamma_S$ is reduced to the two extremities of Γ_S. The momentum balance is

obtained by writing that F in Eq. (18) is given by $F = -\sqrt{1 + \eta_x^2}\,\sigma(u, p)\tilde{n} \cdot n$, where $\tilde{n}(x, y)$ is the unit normal at $(x, y) \in \Gamma_S(t)$ to $\Gamma_S(t)$ outward $\Omega_F(t)$, and n is the unit normal to Γ_S outward $\Omega_F(0) = \Omega_F$. If in addition, a control f acts as a distributed control in the beam equation, we shall have

$$F = -\sqrt{1 + \eta_x^2}\,\sigma(u, p)\tilde{n} \cdot n + f \qquad (19)$$

The equality of velocities on $\Gamma_S(t)$ reads as

$$u(x, y_0 + \eta(x, t)) = (0, \eta_t(x, t)),$$
$$x \in (0, L), t > 0. \qquad (20)$$

Control of Fluid-Structure Models

To control or to stabilize fluid-structure models, the control may act either in the fluid equation or in the structure equation or in both equations. There are a very few controllability and stabilization results for systems coupling the incompressible Navier-Stokes system with a structure equation. We state below two of those results. Some other results are obtained for simplified one-dimensional models coupling the viscous Burgers equation coupled with the motion of a mass; see Badra and Takahashi (2013) and the references therein.

We also have to mention here recent papers on control problems for systems coupling quasi-stationary Stokes equations with the motion of deformable bodies, modeling microorganism swimmers at low Reynolds number; see Alouges et al. (2008).

## Null Controllability of the Navier-Stokes System Coupled with the Motion of a Rigid Body

The system coupling the incompressible Navier-Stokes system Eq. (3) in the domain drawn in Fig. 1, with the motion of a rigid body

described by Eqs. (13)–(16), with the boundary control Eq. (17) is null controllable locally in a neighborhood of 0. Before linearizing the system in a neighborhood of 0, the fluid equations have to be rewritten in Lagrangian coordinates, that is, in the cylindrical domain $\Omega_F \times (0, \infty)$. The linearized system is the Stokes system coupled with a system of ordinary differential equations. The proof of this null controllability result relies on a Carleman estimate for the adjoint system; see, e.g., Boulakia and Guerrero (2013).

Feedback Stabilization of the Navier-Stokes System Coupled with a Beam Equation

The system coupling the incompressible Navier-Stokes system Eq. (3) in the domain drawn in Fig. 2, with beam Eqs. (18)–(20), can be locally stabilized with any prescribed exponential decay rate $-\alpha < 0$, by a feedback control f acting in Eq. (18) via Eq. (19); see Raymond (2010). The proof consists in showing that the infinitesimal generator of the linearized model is an analytic semigroup (when $c > 0$), that its resolvent is compact, and that the Hautus criterion is satisfied.

When the control acts in the fluid equation, the system coupling Eq. (3) in the domain drawn in Fig. 2, with the beam Eqs. (18)–(20), can be stabilized when $c > 0$. To the best of our knowledge, there is no null controllability result for such systems, even with controls acting both in the structure and fluid equations. The case where the beam equation is approximated by a finite-dimensional model is studied in Lequeurre (2013).
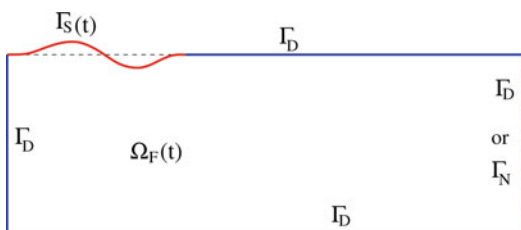
Control of Fluids and Fluid-Structure Interactions, Fig. 2

Cross-References

▶ Bilinear Control of Schrödinger PDEs
▶ Boundary Control of Korteweg-de Vries and Kuramoto–Sivashinsky PDEs
▶ Motion Planning for PDEs

Bibliography

Alouges F, DeSimone A, Lefebvre A (2008) Optimal strokes for low Reynolds number swimmers: an example. J Nonlinear Sci 18:277–302

Badra M (2009) Lyapunov function and local feedback boundary stabilization of the Navier-Stokes equations. SIAM J Control Optim 48:1797–1830

Badra M, Takahashi T (2013) Feedback stabilization of a simplified 1d fluid-particle system. An. IHP, Analyse Non Lin. http://dx.doi.org/10.1016/j.anihpc.2013.03.009

Barbu V, Lasiecka I, Triggiani R (2006) Tangential boundary stabilization of Navier-Stokes equations. Mem Am Math Soc 181 (852) 128

Boulakia M, Guerrero S (2013) Local null controllability of a fluid-solid interaction problem in dimension 3. J Eur Math Soc 15:825–856

Chambolle A, Desjardins B, Esteban MJ, Grandmont C (2005) Existence of weak solutions for unsteady fluid-plate interaction problem. J Math Fluid Mech 7:368–404

Coron J-M (1996) On the controllability of 2-D incompressible perfect fluids. J Math Pures Appl 75(9):155–188

Coron J-M (2007) Control and nonlinearity. American Mathematical Society, Providence

Ervedoza S, Glass O, Guerrero S, Puel J-P (2012) Local exact controllability for the one-dimensional compressible Navier-Stokes equation. Arch Ration Mech Anal 206:189–238

Fabre C, Lebeau G (1996) Prolongement unique des solutions de l'équation de Stokes. Comm. P. D. E. 21:573–596

Fernandez-Cara E, Guerrero S, Imanuvilov Yu O, Puel J-P (2004) Local exact controllability of the Navier-Stokes system. J Math Pures Appl 83:1501–1542

Fursikov AV (2004) Stabilization for the 3D Navier-Stokes system by feedback boundary control. Partial differential equations and applications. Discrete Contin Dyn Syst 10:289–314

Fursikov AV, Imanuvilov Yu O (1996) Controllability of evolution equations. Lecture notes series, vol 34. Seoul National University, Research Institute of Mathematics, Global Analysis Research Center, Seoul

Lequeurre J (2013) Null controllability of a fluid-structure system. SIAM J Control Optim 51:1841–1872

Raymond J-P (2006) Feedback boundary stabilization of the two dimensional Navier-Stokes equations. SIAM J Control Optim 45:790–828

Raymond J-P (2007) Feedback boundary stabilization of the three-dimensional incompressible Navier-Stokes equations. J Math Pures Appl 87:627–669

Raymond J-P (2010) Feedback stabilization of a fluid–structure model. SIAM J Control Optim 48:5398–5443

Raymond J-P, Thevenet L (2010) Boundary feedback stabilization of the two dimensional Navier-Stokes equations with finite dimensional controllers. Discret Contin Dyn Syst A 27:1159–1187

Vazquez R, Krstic M (2008) Control of turbulent and magnetohydrodynamic channel flows: boundary stabilization and estimation. Birkhäuser, Boston

Control of Linear Systems with Delays

Wim Michiels
KU Leuven, Leuven (Heverlee), Belgium

Abstract

The presence of time delays in dynamical systems may induce complex behavior, and this behavior is not always intuitive. Even if a system's equation is scalar, oscillations may occur. Time delays in control loops are usually associated with degradation of performance and robustness, but, at the same time, there are situations where time delays are used as controller parameters.

Keywords

Delay differential equations; Delays as controller parameters; Functional differential equation

Introduction

Time-delays are important components of many systems from engineering, economics, and the life sciences, due to the fact that the transfer of material, energy, and information is mostly not instantaneous. They appear, for instance, as computation and communication lags, they model transport phenomena and heredity, and they arise as feedback delays in control loops. An overview of applications, ranging from traffic flow control and lasers with phase-conjugate feedback, over (bio)chemical reactors and cancer modeling, to control of communication networks and control via networks, is included in Sipahi et al. (2011).

The aim of this contribution is to describe some fundamental properties of linear control systems subjected to time-delays and to outline principles behind analysis and synthesis methods. Throughout the text, the results will be illustrated by means of the scalar system

$$\dot{x}(t) = u(t - \tau), \tag{1}$$

which, controlled with instantaneous state feedback, $u(t) = -kx(t)$, leads to the closed-loop system

$$\dot{x}(t) = -kx(t - \tau). \tag{2}$$

Although this didactic example is extremely simple, we shall see that its dynamics are already very rich and shed a light on delay effects in control loops.

In some works, the analysis of (2) is called the *hot shower problem*, as it can be interpreted as a (over)simplified model for a human adjusting the temperature in a shower: $x(t)$ then denotes the difference between the water temperature and the desired temperature as felt by the person, the term $-kx(t)$ models the reaction of the person by further opening or closing taps, and the delay is due to the propagation with finite speed of the water in the ducts.

Basis Properties of Time-Delay Systems

Functional Differential Equation
We focus on a model for a time-delay system described by

$$\dot{x}(t) = A_0 x(t) + A_1 x(t - \tau), \quad x(t) \in \mathbb{R}^n. \tag{3}$$

This is an example of a *functional differential equation* (FDE) of *retarded type*. The term FDE stems from the property that the right-hand side can be interpreted as a functional evaluated at a piece of trajectory. The term retarded expresses that the right-hand side does not explicitly depend on $\dot{x}$.

As a first difference with an ordinary differential equation, the initial condition of (3) at $t = 0$ is a function ϕ from $[-\tau, 0]$ to $\mathbb{R}^n$. For all $\phi \in \mathcal{C}\left([-\tau, 0], \mathbb{R}^n\right)$, where $\mathcal{C}\left([-\tau, 0], \mathbb{R}^n\right)$ is the space of continuous functions mapping the interval $[-\tau, 0]$ into $\mathbb{R}^n$, a forward solution $x(\phi)$ exists and is uniquely defined. In Fig. 1, a solution of the scalar system (2) is shown.

The discontinuity in the derivative at $t = 0$ stems from $A_0\phi(0) + A_1\phi(-\tau) \neq \lim_{\theta \to 0}\dot{\phi}$. Due to the smoothing property of an integrator, however, at $t = n \in \mathbb{N}$, the discontinuity will only be present in the $(n + 1)$th derivative. This illustrates a second property of functional

differential equations of retarded type: solutions become smoother as time evolves. As a third major difference with ODEs, backward continuation of solutions is not always possible (Michiels and Niculescu 2007).

Reformulation in a First-Order Form

The state of system (3) at time t is the minimal information needed to continue the solution, which, once again, boils down to a function segment $x_t(\phi)$ where $x_t(\phi)(\theta) = x(t + \theta)$, $\theta \in [-\tau, 0]$ (in Fig. 1, the function x_t is shown in red for $t = 5$). This suggests that (3) can be reformulated as a standard ordinary differential equation over the infinite-dimensional space $\mathcal{C}([-\tau, 0], \mathbb{R}^n)$. This equation takes the form

$$\frac{d}{dt}z(t) = \mathcal{A}z(t), \; z(t) \in \mathcal{C}\left([-\tau, 0], \mathbb{R}^n\right) \quad (4)$$

where operator $\mathcal{A}$ is given by

$$\mathcal{D}(\mathcal{A}) = \left\{ \phi \in \mathcal{C}\left([-\tau_m, 0], \mathbb{R}^n\right) : \begin{array}{l} \dot{\phi} \in \mathcal{C}\left([-\tau_m, 0], \mathbb{R}^n\right) \\ \dot{\phi}(0) = A_0\phi(0) + A_1\phi(-\tau) \end{array} \right\},$$

$$\mathcal{A}\phi \qquad\qquad\qquad = \frac{d\phi}{d\theta}. \qquad\qquad\qquad (5)$$

The relation between solutions of (3) and (4) is given by $z(t)(\theta) = x(t + \theta)$, $\theta \in [-\tau, 0]$. Note that all system information is concentrated in the nonlocal boundary condition describing the domain of $\mathcal{A}$. The representation (4) is closely related to a description by an advection PDE with a nonlocal boundary condition (Krstic 2009).

Asymptotic Growth Rate of Solutions and Stability

The reformulation of (3) into the standard form (4) allows us to define stability notions and to generalize the stability theory for ordinary differential equations in a straightforward way, with the main change that the state space is $\mathcal{C}([-\tau, 0], \mathbb{R}^n)$. For example, the null solution of (3) is exponentially stable if and only if there exist constants $C > 0$ and $\gamma > 0$ such that

$$\forall \phi \in \mathcal{C}\left([-\tau_m, 0], \mathbb{R}^n\right) \|x_t(\phi)\|_s \leq Ce^{-\gamma t}\|\phi\|_s,$$

where $\|\cdot\|_s$ is the supremum norm and $\|\phi\|_s = \sup_{\theta \in [-\tau, 0]}\|\phi(\theta)\|_2$. As the system is linear,
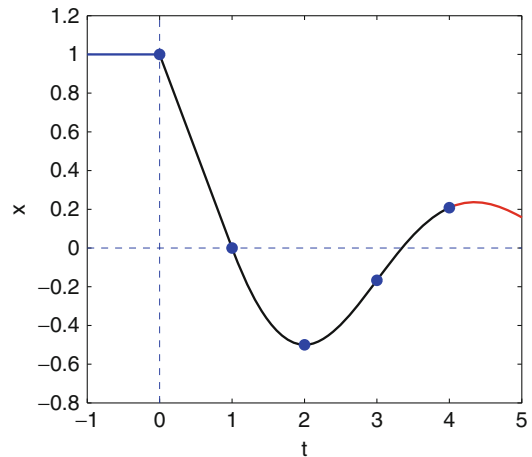
Control of Linear Systems with Delays, Fig. 1 Solution of (2) for $\tau = 1, k = 1$, and initial condition $\phi \equiv 1$

asymptotic stability and exponential stability are equivalent. A direct generalization of Lyapunov's second method yields:

Theorem 1 *The null solution of linear system* (3) *is asymptotically stable if there exist a continuous functional* $V : \mathcal{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}$ *(a so-called Lyapunov-Krasovskii functional) and continuous nondecreasing functions* $u, v, w :$ $\mathbb{R}^+ \to \mathbb{R}^+$ *with*

$$u(0) = v(0) = w(0) = 0 \text{ and } u(s) > 0,$$
$$v(s) > 0, w(s) > \text{ for } s > 0,$$

such that for all $\phi \in \mathcal{C}([-\tau, 0], \mathbb{R}^n)$

$$u(\|\phi\|_s) \le V(\phi) \le v(\|\phi()\|_2),$$
$$\dot{V}(\phi) \le -w(\|\phi()\|_2),$$

where

$$\dot{V}(\phi) = \lim_{h \to 0+} \sup \frac{1}{h}[V(x_h(\phi)) - V(\phi)].$$

Converse Lyapunov theorems and the construction of the so-called complete-type Lyapunov-Krasovskii functionals are discussed in Kharitonov (2013). Imposing a particular structure on the functional, e.g., a form depending only on a finite number of free parameters, often leads to easy-to-check stability criteria (for instance, in the form of LMIs), yet as price to pay, the obtained results may be conservative in the sense that the sufficient stability conditions might not be close to necessary conditions. As an alternative to Lyapunov functionals, Lyapunov functions can be used as well, provided that the condition $\dot{V} < 0$ is relaxed (the so-called Lyapunov-Razumikhin approach); see, for example, Gu et al. (2003).

Delay Differential Equations as Perturbation of ODEs

Many results on stability, robust stability, and control of time-delay systems are explicitly or implicitly based on a perturbation point of view, where delay differential equations are seen as perturbations of ordinary differential equations. For instance, in the literature, a classification of stability criteria is often presented in terms

of *delay-independent* criteria (conditions holding for all values of the delays) and *delay-dependent* criteria (usually holding for all delays smaller than a bound). This classification has its origin at two different ways of seeing (3) as a perturbation of an ODE, with as nominal system $\dot{x}(t) = A_0 x(t)$ and $\dot{x}(t) = (A_0 + A_1)x(t)$ (system for zero delay), respectively. This observation is illustrated in Fig. 2 for results based on input-output- and Lyapunov-based approaches.

The Spectrum of Linear Time-Delay Systems
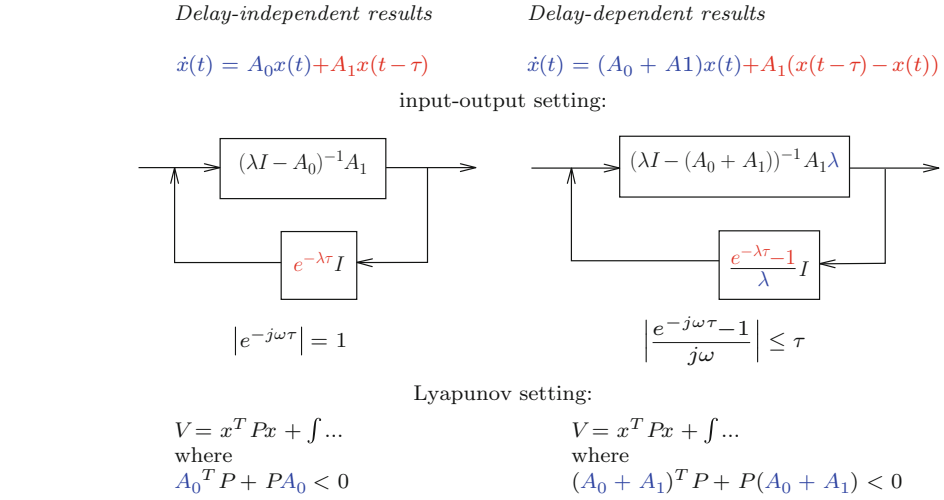
Two Eigenvalue Problems

The substitution of an exponential solution in (3) leads us to the *nonlinear eigenvalue problem*

$$(\lambda I - A_0 - A_1 e^{-\lambda \tau})v = 0, \lambda \in \mathbb{C}, v \in \mathbb{C}^n, v \ne 0. \tag{6}$$

The solutions of the equation $\det(\lambda I - A_0 - A_1 e^{-\lambda \tau}) = 0$ are called characteristic roots. Similarly, formulation (4) leads to the equivalent *infinite-dimensional linear eigenvalue problem*

$$(\lambda I - \mathcal{A})u = 0, \lambda \in \mathbb{C}, u \in \mathcal{C}([-\tau, 0], \mathbb{C}^n), u \ne 0. \tag{7}$$

The combination of these two viewpoints lays at the basis of most methods for computing characteristic roots; see Michiels (2012). On the one hand, discretizing (7), i.e., approximating $\mathcal{A}$ with a matrix, and solving the resulting standard eigenvalue problems allow to obtain global information, for example, estimates of *all* characteristic roots in a given compact set or in a given right half plane. On the other hand, the (finitely many) nonlinear equations (6) allow to make *local corrections* on characteristic root approximations up to the desired accuracy, e.g., using Newton's method or inverse residual iteration. Linear time-delay systems satisfy spectrum-determined growth properties of solutions. For instance, the zero solution of (3) is asymptotically stable if and only if all characteristic roots are in the open left half plane.

Delay-independent results   *Delay-dependent results*

$$\dot{x}(t) = A_0 x(t) + A_1 x(t-\tau) \qquad \dot{x}(t) = (A_0 + A1)x(t) + A_1(x(t-\tau) - x(t))$$

input-output setting:

$$(\lambda I - A_0)^{-1} A_1 \qquad\qquad (\lambda I - (A_0 + A_1))^{-1} A_1 \lambda$$

$$e^{-\lambda\tau} I \qquad\qquad\qquad \frac{e^{-\lambda\tau} - 1}{\lambda} I$$

$$\left| e^{-j\omega\tau} \right| = 1 \qquad\qquad\qquad \left| \frac{e^{-j\omega\tau} - 1}{j\omega} \right| \le \tau$$

Lyapunov setting:

$$V = x^T P x + \int \dots \qquad\qquad V = x^T P x + \int \dots$$
where   where
$$A_0{}^T P + P A_0 < 0 \qquad\qquad (A_0 + A_1)^T P + P(A_0 + A_1) < 0$$

Control of Linear Systems with Delays, Fig. 2 The classification of stability criteria in delay-independent results and delay-dependent results stems from two different perturbation viewpoints. Here, perturbation terms are printed in *red*
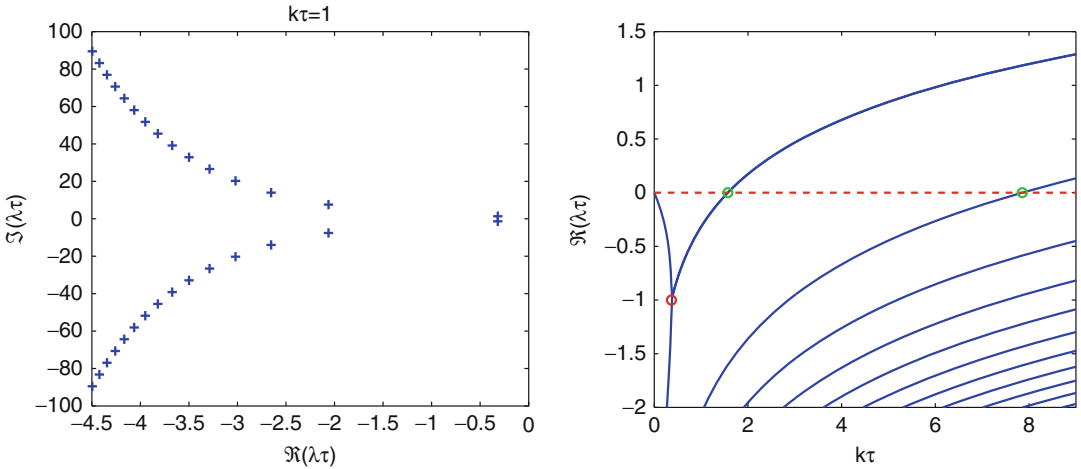
Control of Linear Systems with Delays, Fig. 3 (*Left*) Rightmost characteristic roots of (2) for $k\tau = 1$. (*Right*) Real parts of rightmost characteristic roots as a function of $k\tau$

In Fig. 3 (left), the rightmost characteristic roots of (2) are depicted for $k\tau = 1$. Note that since the characteristic equation can be written as $\lambda\tau + k\tau e^{-\lambda\tau} = 0$, k and τ can be combined into one parameter. In Fig. 3 (right), we show the real parts of the characteristic roots as a function of $k\tau$. The plots illustrate some important spectral properties of retarded-type FDEs. First, even though there are in general infinitely many characteristic roots, the number of them in *any* right half plane is always finite. Second, the individual characteristic roots, as well as the *spectral*

abscissa, i.e., the supremum of the real parts of all characteristic roots, continuously depend on parameters. Related to this, a loss or gain of stability is always associated with characteristic roots crossing the imaginary axis. Figure 3 (right) also illustrates the transition to a delay-free system as $k\tau \to 0^+$.

Critical Delays: A Finite-Dimensional Characterization

Assume that for a given value of k, we are looking for values of the delay τ_c for which (2)

has a characteristic root $j\omega_c$ on the imaginary axis. From $j\omega = -ke^{-j\omega\tau}$, we get

$$\omega_c = k, \ \tau_c = \frac{\frac{\pi}{2} + l2\pi}{\omega_c}, \ l$$

$$= 0, 1, \ldots, \Re\left\{\frac{d\lambda}{d\tau}|_{(\tau_c, j\omega_c)}\right\}^{-1} = \frac{1}{\omega_c^2}. \quad (8)$$

Critical delay values τ_c are indicated with green circles on Fig. 3 (right). The above formulas first illustrate an *invariance property* of imaginary axis roots and their crossing direction with respect to delay shifts of $2\pi/\omega_c$. Second, the number of possible values of ω_c is one and thus *finite*. More generally, substituting $\lambda = j\omega$ in (6) and treating τ as a free parameter lead to a *two-parameter eigenvalue problem*

$$(j\omega I - A_0 - A_1 z)v = 0, \quad (9)$$

with ω on the real axis and $z := \exp(-j\omega\tau)$ on the unit circle. Most methods to solve such a problem boil down to an elimination of one of the independent variables ω or z. As an example of an elimination technique, we directly get from (9)

$$j\omega \in \sigma(A_0 + A_1 z), \ -j\omega \in \sigma(A_0^* + A_1^* z^{-1})$$

$$\Rightarrow \det\left((A_0 + A_1 z) \oplus (A_0^* + A_1^* z^{-1})\right) = 0,$$

where $\sigma(\cdot)$ denotes the spectrum and $\oplus$ the Kronecker sum. Clearly, the resulting eigenvalue problem in z is finite dimensional.

Control of Linear Time-Delay System

Limitations Induced by Delays

It is well known that delays in control loop may lead to a significant degradation of performance and robustness and even to instability (Niculescu 2001; Richard 2003). Let us return to example (2). As illustrated with Fig. 3 and expressions (8), the system loses stability if τ reaches the value $\pi/2k$, while stability cannot be recovered for larger delays. The maximum achievable exponential decay rate of the solutions, which corresponds to the minimum of the spectral abscissa, is given by $-1/\tau$; hence, large delays can only be tolerated at the price of a degradation of the rate of convergence. It should be noted that the limitations induced by delays are even more stringent if the uncontrolled systems are exponentially unstable, which is not the case for (2).

The analysis in the previous sections gives a hint why control is difficult in the presence of delays: the system is inherently infinite dimensional. As a consequence, most control design problems which involve determining a finite number of parameters can be interpreted as reduced-order control design problems or as control design problems for under-actuated systems, which both are known to be hard problems.

Fixed-Order Control

Most standard control design techniques lead to controllers whose dimension is larger or equal to the dimension of the system. For infinite-dimensional time-delay system, such controllers might have a disadvantage of being complicated and hard to implement. To see this, for a system with delay in the state, the generalization of static state feedback, $u(t) = k(x)$, is given by $u(t) = \int_{-\tau}^{0} x(t + \theta)d\mu(\theta)$, where μ is a function of bounded variation. However, in the context of large-scale systems, it is known that reduced-order controllers often perform relatively well compared to full-order controllers, while they are much easier to implement.

Recently, new methods for the design of controllers with a prescribed order (dimension) or structure have been proposed (Michiels 2012). These methods rely on a direct optimization of appropriately defined cost functions (spectral abscissa, $\mathcal{H}_2/\mathcal{H}_\infty$ criteria). While $\mathcal{H}_2$ criteria can be addressed within a derivative-based optimization framework, $\mathcal{H}_\infty$ criteria and the spectral abscissa require targeted methods for *non-smooth optimization problems*. To illustrate the need for such methods, consider again Fig. 3 (right): minimizing the spectral abscissa for a given value of τ as a function of the controller gain k leads to an optimum where the objective function is not differentiable, even not locally Lipschitz, as shown

by the red circle. In case of multiple controller parameters, the path of steepest descent in the parameter space typically has phases along a manifold characterized by the non-differentiability of the objective function.

Using Delays as Controller Parameters

In contrast to the detrimental effects of delays, there are situations where delays have a beneficial effect and are even used as controller parameters; see Sipahi et al. (2011). For instance, delayed feedback can be used to stabilize oscillatory systems where the delay serves to adjust the phase in the control loop. Delayed terms in control laws can also be used to approximate derivatives in the control action. Control laws which depend on the difference $x(t) - x(t - \tau)$, the so-called Pyragas-type feedback, have the property that the position of equilibria and the shape of periodic orbits with period τ are not affected, in contrary to their stability properties. Last but not least, delays can be used in control schemes to generate predictions or to stabilize predictors, which allow to compensate delays and improve performance (Krstic 2009; Zhong 2006). Let us illustrate the main idea once more with system (1).

System (1) has a special structure, in the sense that the delay is only in the input, and it is advantageous to exploit this structure in the context of control. Coming back to the didactic example, the person who is taking a shower is – possibly after some bad experiences – aware about the delay and will take into account his/her prediction of the system's reaction when adjusting the cold and hot water supply. Let us, to conclude, formalize this. The uncontrolled system can be rewritten as $\dot{x}(t) = v(t)$, where $v(t) = u(t - \tau)$. We know u up to the current time t; thus, we know v up to time $t + \tau$, and if $x(t)$ is also known, we can predict the value of x at time $t + \tau$,

$$x_p(t + \tau) = x(t) + \int_t^{t+\tau} v(s)ds$$
$$= x(t) + \int_{t-\tau}^t u(s)ds,$$

and use the predicted state for feedback. With the control law $u(t) = -kx_p(t + \tau)$, there is only one closed-loop characteristic root at $\lambda = -k$, i.e., as long as the model used in the predictor is exact, the delay in the loop is compensated by the prediction. For further reading on prediction-based controllers, see, e.g., Krstic (2009) and the references therein.

Conclusions

Time-delay systems, which appear in a large number of applications, are a class of infinite-dimensional systems, resulting in rich dynamics and challenges from a control point of view. The different representations and interpretations and, in particular, the combination of viewpoints lead to a wide variety of analysis and synthesis tools.

Cross-References

▶ Control of Nonlinear Systems with Delays
▶ H-Infinity Control
▶ H₂ Optimal Control
▶ Optimization-Based Control Design Techniques and Tools

Bibliography

Gu K, Kharitonov VL, Chen J (2003) Stability of time-delay systems. Birkhäuser, Basel

Kharitonov VL (2013) Time-delay systems. Lyapunov functionals and matrices. Birkhäuser, Basel

Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhäuser, Basel

Michiels W (2012) Design of fixed-order stabilizing and $\mathcal{H}_2 - \mathcal{H}_\infty$ optimal controllers: an eigenvalue optimization approach. In: Time-delay systems: methods, applications and new trends. Lecture notes in control and information sciences, vol 423. Springer, Berlin/Heidelberg, pp 201–216

Michiels W, Niculescu S-I (2007) Stability and stabilization of time-delay systems: an eigenvalue based approach. SIAM, Philadelphia

Niculescu S-I (2001) Delay effects on stability: a robust control approach. Lecture notes in control and information sciences, vol 269. Springer, Berlin/New York

Richard J-P (2003) Time-delay systems: an overview of recent advances and open problems. Automatica 39(10):1667–1694

Sipahi R, Niculescu S, Abdallah C, Michiels W, Gu K (2011) Stability and stabilization of systems with time-delay. IEEE Control Syst Mag 31(1):38–65

Zhong Q-C (2006) Robust control of time-delay systems. Springer, London

Control of Machining Processes

Kaan Erkorkmaz
Department of Mechanical & Mechatronics
Engineering, University of Waterloo, Waterloo,
ON, Canada

Abstract

Control of machining processes encompasses a broad range of technologies and innovations, ranging from optimized motion planning and servo drive loop design to on-the-fly regulation of cutting forces and power consumption to applying control strategies for damping out chatter vibrations caused by the interaction of the chip generation mechanism with the machine tool structural dynamics. This article provides a brief introduction to some of the concepts and technologies associated with machining process control.

Keywords

Adaptive control; Chatter vibrations; Feed drive control; Machining; Trajectory planning

Introduction

Machining is used extensively in the manufacturing industry as a shaping process, where high product accuracy, quality, and strength are required. From automotive and aerospace components, to dies and molds, to biomedical implants, and even mobile device chassis, many manufactured products rely on the use of machining.

Machining is carried out on machine tools, which are multi-axis mechatronic systems designed to provide the relative motion between the tool and workpiece, in order to facilitate the desired cutting operation. Figure 1 illustrates a single axis of a ball screw-driven machine tool, performing a milling operation. Here, the cutting process is influenced by the motion of the servo drive. The faster the part is fed in towards the rotating cutter, the larger the cutting forces become, following a typically proportional relationship that holds for a large class of milling operations (Altintas 2012). The generated cutting forces, in turn, are absorbed by the machine tool and feed drive structure. They cause mechanical deformation and may also excite the vibration modes, if their harmonic content is near the structural natural frequencies. This may, depending on the cutting speed and tool and workpiece engagement conditions, lead to forced vibrations or chatter (Altintas 2012).

The disturbance effect of cutting forces is also felt by the servo control loop, consisting of mechanical, electrical, and digital components. This disturbance may result in the degradation of tool positioning accuracy, thereby leading to part errors. Another input that influences the quality achieved in a machining operation is the commanded trajectory. Discontinuous or poorly designed motion commands, with acceleration discontinuity, lead typically to jerky motion, vibrations, and poor surface finish. Beyond motion controller design and trajectory planning, emerging trends in machining process control include regulating, by feedback, various outcomes of the machining process, such as peak resultant cutting force, spindle power consumption, and amplitude of vibrations caused by the machining process. In addition to using actuators and instrumentation already available on a machine tool, such as feed and spindle drives and current sensors, additional devices, such as dynamometers, accelerometers, as well as inertial or piezoelectric actuators, may need to be used in order to achieve the required level of feedback and control injection capability.

Servo Drive Control

Stringent requirements for part quality, typically specified in microns, coupled with disturbance
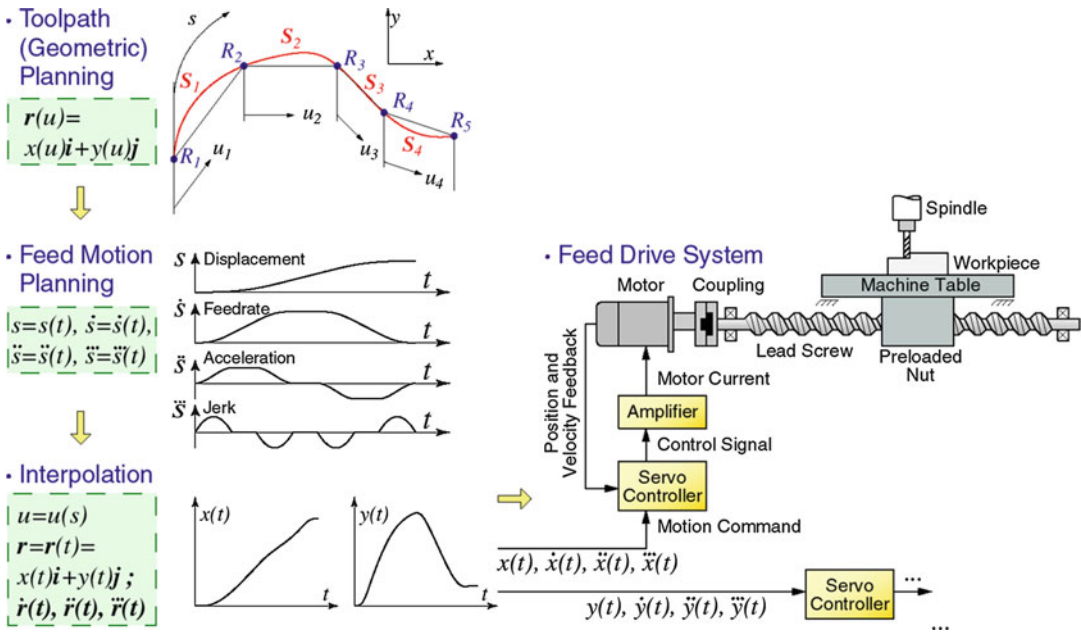
Control of Machining Processes, Fig. 1 Single axis of a ball screw-driven machine tool performing milling

force inputs coming from the machining process, which can be in the order of tens to thousands of Newtons, require that the disturbance rejection of feed drives, which act as dynamic (i.e., frequency dependent) "stiffness" elements, be kept as strong as possible. In traditional machine design, this is achieved by optimizing the mechanical structure for maximum rigidity. Afterwards, the motion control loop is tuned to yield the highest possible bandwidth (i.e., responsive frequency range), without interfering with the vibratory modes of the machine tool in a way that can cause instability. The P-PI position velocity cascade control structure, shown in Fig. 2, is the most widely used technique in machine tool drives. Its tuning guidelines have been well established in the literature (Ellis 2004). To augment the command following accuracy, velocity and acceleration feedforward, and friction compensation terms are added. Increasing the closed-loop bandwidth yields better disturbance rejection and more accurate tracking of the commanded trajectory (Pritschow 1996), which is especially important in high-speed machining applications where elevated cutting speeds necessitate faster feed motion.

It can be seen in Fig. 3 that increased axis tracking errors (e_x and e_y) may result in increased contour error (ε). A practical solution to mitigate this problem, in machine tool engineering, is to also match the dynamics of different motion axes, so that the tracking errors always assume an instantaneous proportion that brings the actual tool position as close as possible to the desired toolpath (Koren 1983). Sometimes, the control action can be designed to directly reduce the contour error as well, which leads to the structure known as "cross-coupling control" (Koren 1980).

Trajectory Planning

Smooth trajectory planning with at least acceleration level continuity is required in machine tool control, in order to avoid inducing unwanted vibration or excessive tracking error during the machining process. For this purpose, computer numerical control (CNC) systems are equipped with various spline toolpath interpolation functions, such as B-splines, and NURBS. The feedrate (i.e., progression speed along the toolpath) is planned in the "look-ahead" function of the

Control of Machining Processes, Fig. 2 P-PI position velocity cascade control used in machine tool drives
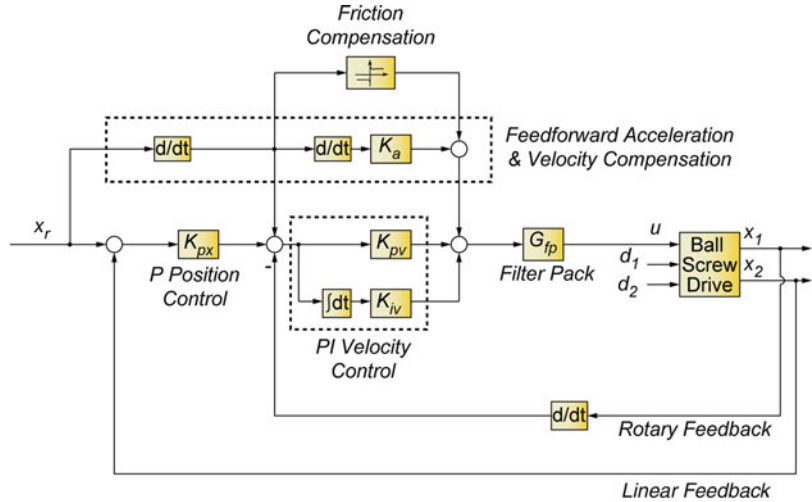
Control of Machining Processes, Fig. 3 Formation of contour error (ε), as a result of servo errors (e_x and e_y) in the individual axes
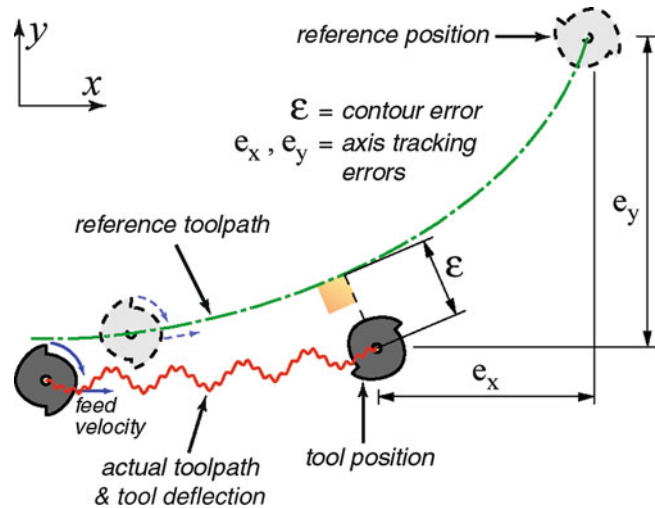

CNC so that the total machining cycle time is reduced as much as possible. This has to be done without violating the position-dependent feedrate limits already programmed into the numerical control (NC) code, which are specified by considering various constraints coming from the machining process.

In feedrate optimization, axis level trajectories have to stay within the velocity and torque limits of the drives, in order to avoid damaging the machine tool or causing actuator saturation. Moreover, as an indirect way of containing tracking errors, the practice of limiting axis level jerk (i.e., rate of change of acceleration) is applied (Gordon and Erkorkmaz 2013). This results in

reduced machining cycle time, while avoiding excessive vibration or positioning error due to "jerky" motion.

An example of trajectory planning using quintic (5th degree) polynomials for toolpath parameterization is shown in Fig. 4. Here, comparison is provided between unoptimized and optimized feedrate profiles subject to the same axis velocity, torque (i.e., control signal), and jerk limits. As can be seen, significant machining time reduction can be achieved through trajectory optimization, while retaining the dynamic tool position accuracy. While Fig. 4 shows the result of an elaborate nonlinear optimization approach (Altintas and Erkorkmaz 2003), practical look-ahead
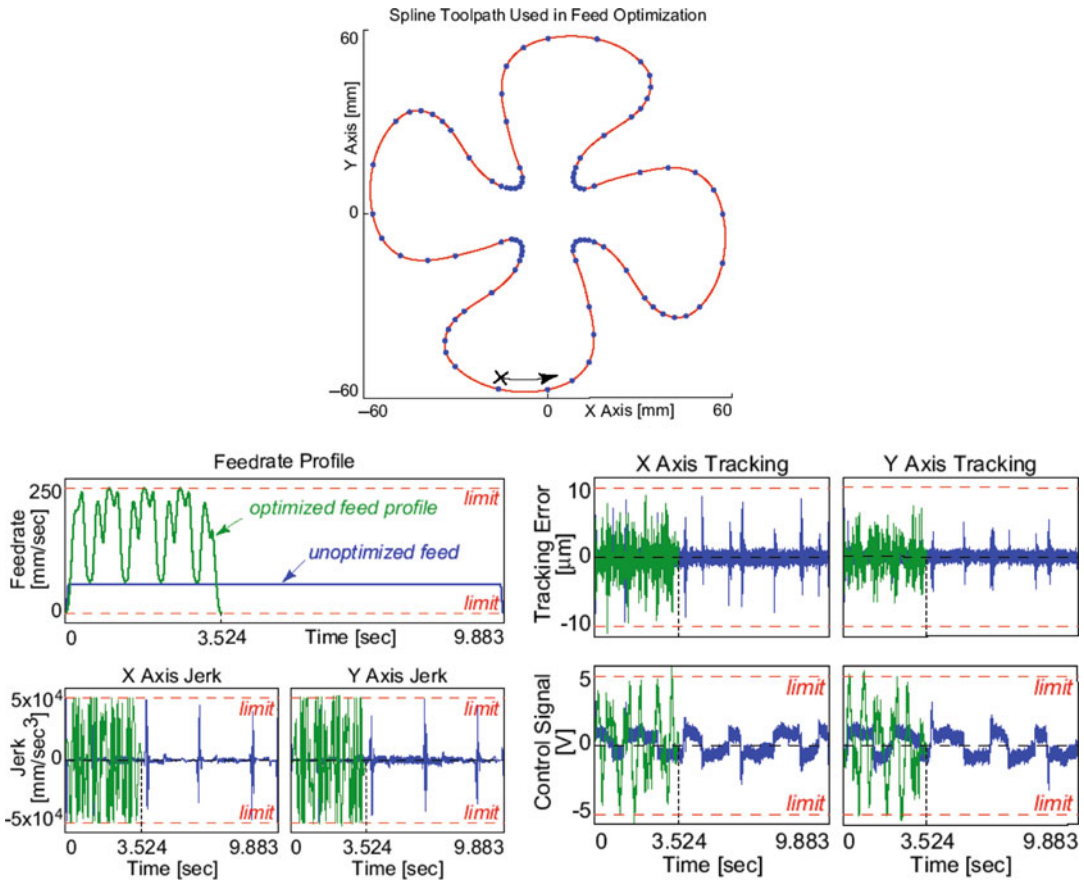
Control of Machining Processes, Fig. 4 Example of quintic spline trajectory planning without and with feedrate optimization

algorithms have also been proposed which lead to more conservative cycle times but are much better suited for real-time implementation inside a CNC (Weck et al. 1999).

Adaptive Control of Machining

There are established mathematical methods for predicting cutting forces, torque, power, and even surface finish for a variety of machining operations like turning, boring, drilling, and milling (Altintas 2012). However, when machining complex components, such as gas turbine impellers, or dies and molds, the tool and workpiece engagement and workpiece geometry undergo continuous change. Hence, it may be difficult to apply such prediction models efficiently, unless

they are fully integrated inside a computer-aided process planning environment, as reported for 3-axis machining by Altintas and Merdol (2007).

An alternative approach, which allows the machining process to take place within safe and efficient operating bounds, is to use feedback from the machine tool during the cutting process. This measurement can be of the cutting forces using a dynamometer or the spindle power consumption. This measurement is then used inside a feedback control loop to override the commanded feedrate value, which has direct impact on the cutting forces and power consumption. This scheme can be used to ensure that the cutting forces do not exceed a certain limit for process safety or to increase the feed when the machining capacity is underutilized, thus boosting productivity. Since the geometry and tool engagement are generally
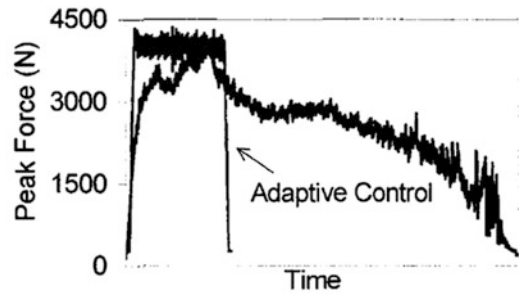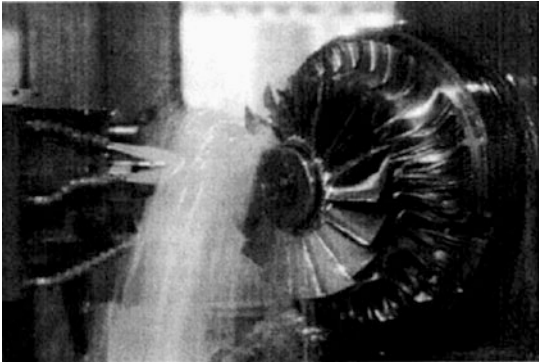
Control of Machining Processes, Fig. 5 Example of 5-axis impeller machining with adaptive force control (Source: Budak and Kops (2000), courtesy of Elsevier)

Control of Machining Processes, Fig. 6 Schematic of the chatter vibration mechanism for one degree of freedom (From: Altintas (2012), courtesy of Cambridge University Press)
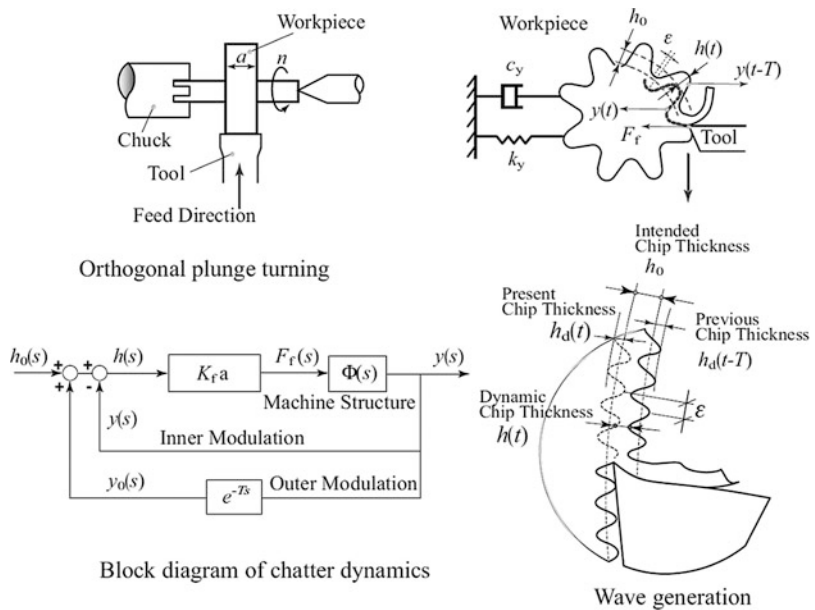
continuously varying, the coefficients of a model that relates the cutting force (or power) to the feed command are also time-varying. Furthermore, in CNC controllers, depending on the trajectory generation architecture, the execution latency of a feed override command may not always be deterministic. Due to these sources of variability, rather than using classical fixed gain feedback, machining control research has evolved around adaptive control techniques (Masory and Koren 1980; Spence and Altintas 1991), where changes in the cutting process dynamics are continuously tracked and the control law, which computes the proceeding feedrate override, is updated

accordingly. This approach has produced significant cycle time reduction in 5-axis machining of gas turbine impellers, as reported in Budak and Kops (2000) and shown in Fig. 5.

Control of Chatter Vibrations

Chatter vibrations are caused by the interaction of the chip generation mechanism with the structural dynamics of the machine, tool, and workpiece assembly (see Fig. 6). The relative vibration between the tool and workpiece generates a wavy surface finish. In the consecutive tool

pass, a new wave pattern, caused by the current instantaneous vibration, is generated on top of the earlier one. If the formed chip, which has an undulated geometry, displays a steady average thickness, then the resulting cutting forces and vibrations also remain bounded. This leads to a stable steady-state cutting regime, known as "forced vibration." On the other hand, if the chip thickness keeps increasing at every tool pass, resulting in increased cutting forces and vibrations, then chatter vibration is encountered. Chatter can be extremely detrimental to the machined part quality, tool life, and the machine tool.

Chatter has been reported in literature to be caused by two main phenomena: self-excitation through regeneration and mode coupling. For further information on chatter theory, the reader is referred to Altintas (2012) as an excellent starting point.

Various mitigation measures have been investigated and proposed in order to avoid and control chatter. One widespread approach is to select chatter-free cutting conditions through detailed modal testing and stability analyses. Recently, to achieve higher material removal rates, the application of active damping has started to receive interest. This has been realized through specially designed tools and actuators (Munoa et al. 2013; Pratt and Nayfeh 2001) and demonstrated productivity improvement in boring and milling operations. As another method for chatter suppression, modulation of the cutting (i.e., spindle) speed has been successfully applied as a means of interrupting the regeneration mechanism (Soliman and Ismail 1997; Zatarain et al. 2008).

Summary and Future Directions

This article has presented an overview of various concepts and emerging technologies in the area of machining process control. The new generation of machine tools, designed to meet the ever-growing productivity and efficiency demands, will likely utilize advanced forms of these ideas and technologies in an integrated manner. As more computational power and better sensors

become available at lower cost, one can expect to see new features, such as more elaborate trajectory planning algorithms, active vibration damping techniques, and real-time process and machine simulation and control capability, beginning to appear in CNC units. No doubt that the dynamic analysis and controller design for such complicated systems will require higher levels of rigor, so that these new technologies can be utilized reliably and at their full potential.

Cross-References

▶ Adaptive Control, Overview
▶ PID Control
▶ Robot Motion Control

Bibliography

Altintas Y (2012) Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design, 2nd edn. Cambridge University Press, Cambridge

Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. Ann CIRP 52(1):297–302

Altintas Y, Merdol DS (2007) Virtual High performance milling. Ann CIRP 55(1):81–84

Budak E, Kops L (2000) Improving productivity and part quality in milling of titanium based impellers by chatter suppression and force control. Ann CIRP 49(1):31–36

Ellis GH (2004) Control system design guide, 3rd edn. Elsevier Academic, New York

Gordon DJ, Erkorkmaz K (2013) Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter. Precis Eng 37(2):308–322

Koren Y (1980) Cross-coupled biaxial computer control for manufacturing systems. ASME J Dyn Syst Meas Control 102:265–272

Koren Y (1983) Computer control of manufacturing systems. McGraw-Hill, New York

Masory O, Koren Y (1980) Adaptive control system for turning. Ann CIRP 29(1):281–284

Munoa J, Mancisidor I, Loix N, Uriarte LG, Barcena R, Zatarain M (2013) Chatter suppression in ram type travelling column milling machines using a biaxial inertial actuator. Ann CIRP 62(1):407–410

Pratt JR, Nayfeh AH (2001) Chatter control and stability analysis of a cantilever boring bar under regenerative cutting conditions. Philos Trans R Soc 359:759–792

Pritschow G (1996) On the influence of the velocity gain factor on the path deviation. Ann CIRP 45/1:367–371

Soliman E, Ismail F (1997) Chatter suppression by adaptive speed modulation. Int J Mach Tools Manuf 37(3):355–369

Spence A, Altintas Y (1991) CAD assisted adaptive control for milling. ASME J Dyn Syst Meas Control 113(3):444–450

Weck M, Meylahn A, Hardebusch C (1999) Innovative algorithms for spline-based CNC controller. Ann Ger Acad Soc Prod Eng VI(1):83–86

Zatarain M, Bediaga I, Munoa J, Lizarralde R (2008) Stability of milling processes with continuous spindle speed variation: analysis in the frequency and time domains, and experimental correlation. Ann CIRP 57(1):379–384

C

Control of Networks of Underwater Vehicles

Naomi Ehrich Leonard
Department of Mechanical and Aerospace
Engineering, Princeton University, Princeton,
NJ, USA

Abstract

Control of networks of underwater vehicles is critical to underwater exploration, mapping, search, and surveillance in the multiscale, spatiotemporal dynamics of oceans, lakes, and rivers. Control methodologies have been derived for tasks including feature tracking and adaptive sampling and have been successfully demonstrated in the field despite the severe challenges of underwater operations.

Keywords

Adaptive sampling; Feature tracking; Gliders; Mobile sensor arrays; Underwater exploration

Introduction

The development of theory and methodology for control of networks of underwater vehicles is motivated by a multitude of underwater applications and by the unique challenges associated with operating in the oceans, lakes, and rivers. Tasks include underwater exploration, mapping, search, and surveillance, associated with problems that include pollution monitoring, human safety, resource seeking, ocean science, and marine archeology. Vehicle networks collect data on underwater physics, biology, chemistry, and geology for improving the understanding and predictive modeling of natural dynamics and human-influenced changes in marine environments. Because the underwater environment is opaque, inhospitable, uncertain, and dynamic, control is critical to the performance of vehicle networks.

Underwater vehicles typically carry sensors to measure external environmental signals and fields, and thus a vehicle network can be regarded as a mobile sensor array. The underlying principle of control of networks of underwater vehicles leverages their mobility and uses an interacting dynamic among the vehicles to yield a high-performing collective behavior. If the vehicles can communicate their state or measure the relative state of others, then they can cooperate and coordinate their motion.

One of the major drivers of control of underwater mobile sensor networks is the multiscale, spatiotemporal dynamics of the environmental fields and signals. In Curtin et al. (1993), the concept of the autonomous oceanographic sampling network (AOSN), featuring a network of underwater vehicles, was introduced for dynamic measurement of the ocean environment and resolution of spatial and temporal gradients in the sampled fields. For example, to understand the coupled biological and physical dynamics of the ocean, data are required both on the small-scale dynamics of phytoplankton, which are major actors in the marine ecosystem and the global climate, and on the large-scale dynamics of the flow field, temperature, and salinity.

Accordingly, control laws are needed to coordinate the motion of networks of underwater vehicles to match the many relevant spatial and temporal scales. And for a network of underwater vehicles to perform complex missions reliably and efficiently, the control must address the many

uncertainties and real-world constraints including the influence of currents on the motion of the vehicles and the limitations on underwater communication.

Vehicles

Control of networks of underwater vehicles is made possible with the availability of small (e.g., 1.5–2 m long), relatively inexpensive autonomous underwater vehicles (AUVs). Propelled AUVs such as the REMUS provide maneuverability and speed. These kinds of AUVs respond quickly and agilely to the needs of the network, and because of their speed, they can often power through strong ocean flows. However, propelled AUVs are limited by their batteries; for extended missions, they need docking stations or other means to recharge their batteries.

Buoyancy-driven autonomous underwater gliders, including the Slocum, the Spray, and the Seaglider, are a class of endurance AUVs designed explicitly for collecting data over large three-dimensional volumes continuously over periods of weeks or even months (Rudnick et al. 2004). They move slowly and steadily, and, as a result, they are particularly well suited to network missions of long duration.

Gliders propel themselves by alternately increasing and decreasing their buoyancy using either a hydraulic or a mechanical buoyancy engine. Lift generated by flow over fixed wings converts the vertical ascent/descent induced by the change in buoyancy into forward motion, resulting in a sawtooth-like trajectory in the vertical plane. Gliders can actively redistribute internal mass to control attitude, for example, they pitch by sliding their battery pack forward and aft. For heading control, they shift mass to roll, bank, and turn or deflect a rudder. Some gliders are designed for deep water, e.g., to 1,500 m, while others for shallower water, e.g., to 200 m.

Gliders are typically operated at their maximum speed and thus they move at approximately constant speed relative to the flow. Because this is relatively slow, on the order of 0.3–0.5 m/s in the horizontal direction and 0.2 m/s in the vertical,

ocean currents can sometimes reach or even exceed the speed of the gliders. Unlike a propelled AUV, which typically has sufficient thrust to maintain course despite currents, a glider trying to move in the direction of a strong current will make no forward progress. This makes coordinated control of gliders challenging; for instance, two sensors that should stay sufficiently far apart may be pushed toward each other leading to less than ideal sampling conditions.

Communication and Sensing

Underwater communication is one of the biggest challenges to the control of networks of underwater vehicles and one that distinguishes it from control of vehicles on land or in the air. Radio-frequency communication is not typically available underwater, and acoustic data telemetry has limitations including sensitivity to ambient noise, unpredictable propagation, limited bandwidth, and latency.

When acoustic communication is too limiting, vehicles can surface periodically and communicate via satellite. This method may be bandwidth limited and will require time and energy. However, in the case of profiling propelled AUVs or underwater gliders, they already move in the vertical plane in a sawtooth pattern and thus regularly come closer to the surface. When on the surface, vehicles can also get a GPS fix whereas there is no access to GPS underwater. The GPS fix is used for correcting onboard dead reckoning of the vehicle's absolute position and for updating onboard estimation of the underwater currents, both helpful for control.

Vehicles are typically equipped with conductivity-temperature-density (CTD) sensors to measure temperature, salinity, and density. From this pressure can be computed and thus depth and vertical speed. Attitude sensors provide measurements of pitch, roll, and heading. Position and velocity in the plane is estimated using dead reckoning. Many sensors for measuring the environment have been developed for use on underwater vehicles; these include chlorophyll fluorometers to estimate phytoplankton abundance, acoustic Doppler

profilers (ADPs) to measure variations in water velocity, and sensors to measure pH, dissolved oxygen, and carbon dioxide.

Control

Described here are a selection of control methodologies designed to serve a variety of underwater applications and to address many of the challenges described above for both propelled AUVs and underwater gliders. Some of these methodologies have been successfully field tested in the ocean.

Formations for Tracking Gradients, Boundaries, and Level Sets in Sampled Fields

While a small underwater vehicle can take only single-point measurements of a field, a network of N vehicles employing cooperative control laws can move as a formation and estimate or track a gradient in the field. This can be done in a straightforward way in 2D with three vehicles and can be extended to 3D with additional vehicles. Consider $N = 3$ vehicles moving together in an equilateral triangular formation and sampling a 2D field $T : \mathbb{R}^2 \to \mathbb{R}$. The formation serves as a sensor array and the triangle side length defines the resolution of the array.

Let the position of the ith vehicle be $\mathbf{x}_i \in \mathbb{R}^2$. Consider double integrator dynamics $\ddot{\mathbf{x}}_i = \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{R}^2$ is the control force on the ith vehicle. Suppose that each vehicle can measure the relative position of each of its neighbors, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Decentralized control that derives from an artificial potential is a popular method for each of the three vehicles to stay in the triangular formation of prescribed resolution d_0. Consider the nonlinear interaction potential $V_I : \mathbb{R}^2 \to \mathbb{R}$ defined as

$$V_I(\mathbf{x}_{ij}) = k_s \left(\ln \|\mathbf{x}_{ij}\| + \frac{d_0}{\|\mathbf{x}_{ij}\|^2} \right)$$

where $k_s > 0$ is a scalar gain. The control law for the ith vehicle derives as the gradient of this potential with respect to $\mathbf{x}_i$ as follows:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i = - \sum_{j=1, j \neq i}^{N} \nabla V_I(\mathbf{x}_{ij}) - k_d \dot{\mathbf{x}}_i$$

where a damping term is added with scalar gain $k_d > 0$. Stability of the triangle of resolution d_0 is proved with the Lyapunov function

$$V = \frac{1}{2} \sum_{i=1}^{N} \|\dot{\mathbf{x}}_i\|^2 + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} V_I(\mathbf{x}_{ij}).$$

Now let each vehicle use the sequence of single-point measurements it takes along its path to compute the projection of the spatial gradient onto its normalized velocity, $\mathbf{e}_{\dot{\mathbf{x}}} = \dot{\mathbf{x}}_i / \|\dot{\mathbf{x}}_i\|$, i.e., $\nabla T_P(\mathbf{x}, \dot{\mathbf{x}}_i) = (\nabla T(\mathbf{x}) \cdot \mathbf{e}_{\dot{\mathbf{x}}}) \mathbf{e}_{\dot{\mathbf{x}}}$. Following Bachmayer and Leonard (2002), let

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i = \kappa \nabla T_P(\mathbf{x}, \dot{\mathbf{x}}_i) - \sum_{j=1, j \neq i}^{N} \nabla V_I(\mathbf{x}_{ij}) - k_d \dot{\mathbf{x}}_i,$$

where κ is a scalar gain. For $\kappa > 0$, each vehicle will accelerate along its path when it measures an increasing T and decelerates for a decreasing T. Each vehicle will also turn to keep up with the others so that the formation will climb the spatial gradient of T to find a local maximum.

Alternative control strategies have been developed that add versatility in feature tracking. The virtual body and artificial potential (VBAP) multivehicle control methodology (Ögren et al. 2004) was demonstrated with a network of Slocum autonomous underwater gliders in the AOSN II field experiment in Monterey Bay, California, in August 2003 (Fiorelli et al. 2006). VBAP is well suited to the operational scenario described above in which vehicles surface asynchronously to establish communication with a base.

VBAP is a control methodology for coordinating the translation, rotation, and dilation of a group of vehicles. A virtual body is defined by a set of reference points that move according to dynamics that are computed centrally and made available to the vehicles in the group. Artificial potentials are used to couple the dynamics of vehicles and a virtual body so that control laws can be derived that stabilize desired formations of vehicles and a virtual body. When sampled

measurements of a scalar field can be communicated, the local gradients can be estimated. Gradient climbing algorithms prescribe virtual body direction, so that, for example, the vehicle network can be directed to head for the coldest water or the highest concentration of phytoplankton. Further, the formation can be dilated so that the resolution can be adapted to minimize error in estimates. Control of the speed of the virtual body ensures stability and convergence of the vehicle formation.

These ideas have been extended further to design provable control laws for cooperative level set tracking, whereby small vehicle groups cooperate to generate contour plots of noisy, unknown fields, adjusting their formation shape to provide optimal filtering of their noisy measurements (Zhang and Leonard 2010).

Motion Patterns for Adaptive Sampling

A central objective in many underwater applications is to design provable and reliable mobile sensor networks for collecting the richest data set in an uncertain environment given limited resources. Consider the sampling of a single time- and space-varying scalar field, like temperature T, using a network of vehicles, where the control problem is to coordinate the motion of the network to maximize information on this field over a given area or volume.

The definition of the information metric will depend on the application. If the data are to be assimilated into a high-resolution dynamical ocean model, then the metric would be defined by uncertainty as computed by the model. A general-purpose metric, based on objective analysis (linear statistical estimation from given field statistics), specifies the statistical uncertainty of the field model as a function of where and when the data were taken (Bennett 2002). The posteriori error $A(\mathbf{r}, t)$ is the variance of T about its estimate at location $\mathbf{r}$ and time t. Entropic information over a spatial domain of area $\mathcal{A}$ is

$$\mathcal{I}(t) = -\log\left(\frac{1}{\sigma_0 \mathcal{A}} \int d\mathbf{r}\, A(\mathbf{r}, t)\right),$$

where σ_0 is a scaling factor (Grocholsky 2002).

Computing coordinated trajectories to maximize $\mathcal{I}(t)$ can in principle be addressed using optimal coverage control methods. However, this coverage problem is especially challenging since the uncertainty field is spatially nonuniform and it changes with time and with the motion of the sampling vehicles. Furthermore, the optimal trajectories may become quite complex so that controlling vehicles to them in the presence of dynamic disturbances and uncertainty may lead to suboptimal performance.

An alternative approach decouples the design of motion patterns to optimize the entropic information metric from the decentralized control laws that stabilize the network onto the motion patterns (see Leonard et al. 2007). This approach was demonstrated with a network of 6 Slocum autonomous underwater gliders in a 24-day-long field experiment in Monterey Bay, California, in August 2006 (see Leonard et al. 2010). The coordinating feedback laws for the individual vehicles derive systematically from a control methodology that provides provable stabilization of a parameterized family of collective motion patterns (Sepulchre et al. 2008). These patterns consist of vehicles moving on a finite set of closed curves with spacing between vehicles defined by a small number of "synchrony" parameters. The feedback laws that stabilize a given motion pattern use the same synchrony parameters that distinguish the desired pattern.

Each vehicle moves in response to the relative position and direction of its neighbors so that it keeps moving, it maintains the desired spacing, and it stays close to its assigned curve. It has been observed in the ocean, for vehicles carrying out this coordinated control law, that "when a vehicle on a curve is slowed down by a strong opposing flow field, it will cut inside a curve to make up distance and its neighbor on the same curve will cut outside the curve so that it does not overtake the slower vehicle and compromise the desired spacing" (Leonard et al. 2010). The approach is robust to vehicle failure since there are no leaders in the network, and it is scalable since the control law for each vehicle can be defined in terms of the state of a few other vehicles, independent of the total number of vehicles.

The control methodology prescribes steering laws for vehicles operated at a constant speed. Assume that the ith vehicle moves at unit speed in the plane in the direction $\theta_i(t)$ at time t. Then, the velocity of the ith vehicle is $\dot{\mathbf{x}}_i = (\cos\theta_i, \sin\theta_i)$. The steering control u_i is the component of the force in the direction normal to velocity, such that $\dot{\theta}_i = u_i$ for $i = 1, \ldots, N$. Define

$$U(\theta_1, \ldots, \theta_N) = \frac{N}{2}\|\mathbf{p}_\theta\|^2, \quad \mathbf{p}_\theta = \frac{1}{N}\sum_{j=1}^{N}\dot{\mathbf{x}}_j.$$

U is a potential function that is maximal at 1 when all vehicle directions are synchronized and minimal at 0 when all vehicle directions are perfectly anti-synchronized. Let $\tilde{\mathbf{x}}_i = (\tilde{x}_i, \tilde{y}_i) = (1/N)\sum_{j=1}^{N}\mathbf{x}_{ij}$ and let $\tilde{\mathbf{x}}_i^\perp = (-\tilde{y}_i, \tilde{x}_i)$. Define

$$S(\mathbf{x}_1, \ldots, \mathbf{x}_N, \theta_1, \ldots, \theta_N) = \frac{1}{2}\sum_{i=1}^{N}\|\dot{\mathbf{x}}_i - \omega_0\tilde{\mathbf{x}}_i^\perp\|^2,$$

where $\omega_0 \neq 0$. S is a potential function that is minimal at 0 for circular motion of the vehicles around their center of mass with radius $\rho_0 = |\omega_0|^{-1}$.

Define the steering control as

$$\dot{\theta}_i = \omega_0(1 + K_c\langle\tilde{\mathbf{x}}_i, \dot{\mathbf{x}}_i\rangle) - K_\theta\sum_{j=1}^{N}\sin(\theta_j - \theta_i),$$

where $K_c > 0$ and K_θ are scalar gains. Then, circular motion of the network is a steady solution, with the phase-locked heading arrangement a minimum of $K_\theta U$, i.e., synchronized or perfectly anti-synchronized depending on the sign of K_θ. Stability can be proved with the Lyapunov function $V_{c\theta} = K_c S + K_\theta U$. This steering control law depends only on relative position and relative heading measurements of the other vehicles.

The general form of the methodology extends the above control law to network interconnections defined by possibly time-varying graphs with limited sensing or communication links, and it provides systematic control laws to stabilize symmetric patterns of heading distributions about noncircular closed curves. It also allows

for multiple graphs to handle multiple scales. For example, in the 2006 field experiment, the default motion pattern was one in which six gliders moved in coordinated pairs around three closed curves; one graph defined the smaller-scale coordination of each pair of gliders about its curve, while a second graph defined the larger-scale coordination of gliders across the three curves.

Implementation

Implementation of control of networks of underwater vehicles requires coping with the remote, hostile underwater environment. The control methodology for motion patterns and adaptive sampling, described above, was implemented in the field using a customized software infrastructure called the Glider Coordinated Control System (GCCS) (Paley et al. 2008). The GCCS combines a simple model for control planning with a detailed model of glider dynamics to accommodate the constant speed of gliders, relatively large ocean currents, waypoint tracking routines, communication only when gliders surface (asynchronously), other latencies, and more. Other approaches consider control design in the presence of a flow field, formal methods to integrate high-resolution models of the flow field, and design tailored to propelled AUVs.

Summary and Future Directions

The multiscale, spatiotemporal dynamics of the underwater environment drive the need for well-coordinated control of networks of underwater vehicles that can manage the significant operational challenges of the opaque, uncertain, inhospitable, and dynamic oceans, lakes, and rivers. Control theory and algorithms have been developed to enable networks of vehicles to successfully operate as adaptable sensor arrays in missions that include feature tracking and adaptive sampling. Future work will improve control in the presence of strong and unpredictable flow

fields and will leverage the latest in battery and underwater communication technologies. Hybrid vehicles and heterogeneous networks of vehicles will also promote advances in control. Future work will draw inspiration from the rapidly growing literature in decentralized cooperative control strategies and complex dynamic networks. Dynamics of decision-making teams of robotic vehicles and humans is yet another important direction of research that will impact the success of control of networks of underwater vehicles.

Cross-References

▶ Motion Planning for Marine Control Systems
▶ Underactuated Marine Control Systems

Recommended Reading

In Bellingham and Rajan (2007), it is argued that cooperative control of robotic vehicles is especially useful for exploration in remote and hostile environments such as the deep ocean. A recent survey of robotics for environmental monitoring, including a discussion of cooperative systems, is provided in Dunbabin and Marques (2012). A survey of work on cooperative underwater vehicles is provided in Redfield (2013).

Bibliography

Bachmayer R, Leonard NE (2002) Vehicle networks for gradient descent in a sampled environment. In: Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, pp 112–117

Bellingham JG, Rajan K (2007) Robotics in remote and hostile environments. Science 318(5853):1098–1102

Bennett A (2002) Inverse modeling of the ocean and atmosphere. Cambridge University Press, Cambridge

Curtin TB, Bellingham JG, Catipovic J, Webb D (1993) Autonomous oceanographic sampling networks. Oceanography 6(3):86–94

Dunbabin M, Marques L (2012) Robots for environmental monitoring: significant advancements and applications. IEEE Robot Autom Mag 19(1):24–39

Fiorelli E, Leonard NE, Bhatta P, Paley D, Bachmayer R, Fratantoni DM (2006) Multi-AUV control and adaptive sampling in Monterey Bay. IEEE J Ocean Eng 31(4):935–948

Grocholsky B (2002) Information-theoretic control of multiple sensor platforms. PhD thesis, University of Sydney

Leonard NE, Paley DA, Lekien F, Sepulchre R, Fratantoni DM, Davis RE (2007) Collective motion, sensor networks, and ocean sampling. Proc IEEE 95(1):48–74

Leonard NE, Paley DA, Davis RE, Fratantoni DM, Lekien F, Zhang F (2010) Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. J Field Robot 27(6):718–740

Ögren P, Fiorelli E, Leonard NE (2004) Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment. IEEE Trans Autom Control 49(8):1292–1302

Paley D, Zhang F, Leonard NE (2008) Cooperative control for ocean sampling: the glider coordinated control system. IEEE Trans Control Syst Technol 16(4):735–744

Redfield S (2013) Cooperation between underwater vehicles. In: Seto ML (ed) Marine robot autonomy. Springer, New York, pp 257–286

Rudnick D, Davis R, Eriksen C, Fratantoni D, Perry M (2004) Underwater gliders for ocean research. Mar Technol Soc J 38(1):48–59

Sepulchre R, Paley DA, Leonard NE (2008) Stabilization of planar collective motion with limited communication. IEEE Trans Autom Control 53(3):706–719

Zhang F, Leonard NE (2010) Cooperative filters and control for cooperative exploration. IEEE Trans Autom Control 55(3):650–663

Control of Nonlinear Systems with Delays

Nikolaos Bekiaris-Liberis and Miroslav Krstic
Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Abstract

The reader is introduced to the predictor feedback method for the control of general nonlinear systems with input delays of arbitrary length. The delays need not necessarily be constant but can be time-varying or state-dependent. The predictor feedback methodology employs a model-based construction of the (unmeasurable) future state of

the system. The analysis methodology is based on the concept of infinite-dimensional backstepping transformation – a transformation that converts the overall feedback system to a new, cascade "target system" whose stability can be studied with the construction of a Lyapunov function.

Keywords

Distributed parameter systems; Delay systems; Backstepping; Lyapunov function

Nonlinear Systems with Input Delay

Nonlinear systems of the form

$$\dot{X}(t) = f\left(X(t), U\left(t - D\left(t, X(t)\right)\right)\right), \quad (1)$$

where $t \in \mathbb{R}_+$ is time, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ is a vector field, $X \in \mathbb{R}^n$ is the state, $D : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}_+$ is a nonnegative function of the state of the system, and $U \in \mathbb{R}$ is the scalar input, are ubiquitous in applications. The starting point for designing a control law for (1), as well as for analyzing the dynamics of (1) is to consider the delay-free counterpart of (1), i.e., when $D = 0$, for which a plethora of results exists dealing with its stabilization and Lyapunov-based analysis (Krstic et al 1995).

Systems of the form (1) constitute more realistic models for physical systems than delay-free systems. The reason is that often in engineering applications the control that is applied to the system does not immediately affect the system. This dead time until the controller can affect the system might be due to, among other things, the long distance of the controller from the system, such as, for example, in networked control systems, or due to finite-speed transport or flow phenomena, such as, for example, in additive manufacturing and cooling systems, or due to various after-effects, such as, for example, in population dynamics.

The first step toward control design and analysis for system (1) is to consider the special case in which $D = $ const. The next step is to consider the special case of system (1), in which $D = D(t)$, i.e., the delay is an a priori given function of time. Systems with time-varying delays model numerous real-world systems, such as, networked control systems, traffic systems, or irrigation channels. Assuming that the input delay is an a priori defined function of time is a plausible assumption for some applications. Yet, the time-variation of the delay might be the result of the variation of a physical quantity that has its own dynamics, such as, for example, in milling processes (due to speed variations), 3D printers (due to distance variations), cooling systems (due to flow rate variations), and population dynamics (due to population's size variations). Processes in this category can be modeled by systems with a delay that is a function of the state of the system, i.e., by (1) with $D = D(X)$.

In this article control designs are presented for the stabilization of nonlinear systems with input delays, with delays that are constant (Krstic 2009), time-varying (Bekiaris-Liberis and Krstic 2012) or state-dependent (Bekiaris-Liberis and Krstic 2013b), employing predictor feedback, i.e., employing a feedback law that uses the future rather than the current state of the system. Since one employs in the feedback law the future values of the state, the predictor feedback completely cancels (compensates) the input delay, i.e., after the control signal reaches the system, the state evolves as if there were no delay at all. Since the future values of the state are not a priori known, the main control challenge is the implementation of the predictor feedback law. Having determined the predictor, the control law is then obtained by replacing the current state in a nominal state-feedback law (which stabilizes the delay-free system) by the predictor.

A methodology is presented in the article for the stability analysis of the closed-loop system under predictor feedback by constructing Lyapunov functionals. The Lyapunov functionals are constructed for a transformed (rather than

the original) system. The transformed system is, in turn, constructed by transforming the original actuator state $U(\theta)$, $\theta \in [t - D, t]$ to a transformed actuator state with the aid of an infinite-dimensional backstepping transformation. The overall transformed system is easier to analyze than the original system because it is a cascade, rather than a feedback system, consisting of a delay line with zero input, whose effect fades away in finite time, namely, after D time units, cascaded with an asymptotically stable system.

Predictor Feedback

The predictor feedback designs are based on a feedback law $U(t) = \kappa(X(t))$ that renders the closed-loop system $\dot{X} = f(X, \kappa(X))$ globally asymptotically stable. For stabilizing system (1), the following control law is employed instead

$$U(t) = \kappa(P(t)), \tag{2}$$

where

$$P(\theta) = X(t) + \int_{t-D(t,X(t))}^{\theta} \frac{f(P(s), U(s))}{1 - D_t(\sigma(s), P(s)) - \nabla D(\sigma(s), P(s)) f(P(s), U(s))} ds \tag{3}$$

$$\sigma(\theta) = t + \int_{t-D(t,X(t))}^{\theta} \frac{1}{1 - D_t(\sigma(s), P(s)) - \nabla D(\sigma(s), P(s)) f(P(s), U(s))} ds, \tag{4}$$

for all $t - D(t, X(t)) \leq \theta \leq t$. The signal P is the predictor of X at the appropriate prediction time σ, i.e., $P(t) = X(\sigma(t))$. This fact is explained in more detail in the next paragraphs of this section. The predictor employs the future values of the state X which are not a priori available. Therefore, for actually implementing the feedback law (2) one has to employ (3). Relation (3) is a formula for the future values of the state that depends on the available measured quantities, i.e., the current state $X(t)$ and the history of the actuator state $U(\theta)$, $\theta \in [t - D(t, X(t)), t]$. To make clear the definitions of the predictor P and the prediction time σ, as well as their implementation through formulas (3) and (4), the constant delay case is discussed first.

The idea of predictor feedback is to employ in the control law the future values of the state at the appropriate future time, such that the effect of the input delay is completely canceled (compensated). Define the quantity $\phi(t) = t - D$, which from now on is referred to as the delayed time. This is the time instant at which the control signal that currently affects the system

was actually applied. To cancel the effect of this delay, the control law (2) is designed such that $U(\phi(t)) = U(t - D) = \kappa(X(t))$, i.e., such that $U(t) = \kappa(X(\phi^{-1}(t))) = \kappa(X(t + D))$. Define the prediction time σ through the relation $\phi^{-1}(t) = \sigma(t) = t + D$. This is the time instant at which an input signal that is currently applied actually affects the system. In the case of a constant delay, the prediction time is simply D time-units in the future. Next an implementable formula for $X(\sigma(t)) = X(t + D)$ is derived. Performing a change of variables $t = \theta + D$, for all $t - D \leq \theta \leq t$ in $\dot{X}(t) = f(X(t), U(t - D))$ and integrating in θ starting at $\theta = t - D$, one can conclude that P defined by (3) with $D_t = \nabla D f = 0$ and $D = $ const is the D time-units ahead predictor of X, i.e., $P(t) = X(\sigma(t)) = X(t + D)$.

To better understand definition (3) the case of a linear system with a constant input delay D, i.e., a system of the form $\dot{X}(t) = AX(t) + BU(t - D)$, is considered next (see also ► Control of Linear Systems with Delays and Hale and Verduyn Lunel (1993)). In this case, the predictor $P(t)$ is given explicitly

using the variation of constants formula, with the initial condition $P(t - D) = X(t)$, as $P(t) = e^{AD}X(t) + \int_{t-D}^{t} e^{A(t-\theta)}BU(\theta)d\theta$. For systems that are nonlinear, $P(t)$ cannot be written explicitly, for the same reason that a nonlinear ODE cannot be solved explicitly. So $P(t)$ is represented implicitly using the nonlinear integral equation (3). The computation of $P(t)$ from (3) is straightforward with a discretized implementation in which $P(t)$ is assigned values based on the right-hand side of (3), which involves earlier values of P and the values of the input U.

The case $D = D(t)$ is considered next. As in the case of constant delays the main goal is to implement the predictor P. One needs first to define the appropriate time interval over which the predictor of the state is needed, which, in the constant delay case is simply D time-units in the future. The control law has to satisfy $U(\phi(t)) = \kappa(X(t))$, or, $U(t) = \kappa(X(\sigma(t)))$. Hence, one needs to find an implementable formula for $P(t) = X(\sigma(t))$. In the constant delay case the prediction horizon over which one needs to compute the predictor can be determined based on the knowledge of the delay time since the prediction horizon and the delay time are both equal to D. This is not anymore true in the time-varying case in which the delayed time is defined as $\phi(t) = t - D(t)$, whereas the prediction time as $\phi^{-1}(t) = \sigma(t) = t + D(\sigma(t))$. Employing a change of variables in $\dot{X}(t) = f(X(t), U(t - D(t)))$ as $t = \sigma(\theta)$, for all $\phi(t) \leq \theta \leq t$ and integrating in θ starting at $\theta = \phi(t)$ one obtains the formula for P given by (3) with $D_t = D'(\sigma(t))$, $\nabla Df = 0$ and $D = D(t)$.

Next the case $D = D(X(t))$ is considered. First one has to determine the predictor, i.e., the signal P such that $P(t) = X(\sigma(t))$, where $\sigma(t) = \phi^{-1}(t)$ and $\phi(t) = t - D(X(t))$. In the case of state-dependent delay, the prediction time $\sigma(t)$ depends on the predictor itself, i.e., the time when the current control reaches the system depends on the value of the state at that time, namely, the following implicit relationship holds $P(t) = X(t + D(P(t)))$

(and $X(t) = P(t - D(X(t)))$). This implicit relation can be solved by proceeding as in the time-varying case, i.e., by performing the change of variables $t = \sigma(\theta)$, for all $t - D(X(t)) \leq \theta \leq t$ in $\dot{X}(t) = f(X(t), U(t - D(X(t))))$ and integrating in θ starting at $\theta = t - D(X(t))$, to obtain the formula (3) for P with $D_t = 0$, $\nabla Df = \nabla D(P(s))f(P(s), U(s))$ and $D = D(X(t))$.

Analogously, one can derive the predictor for the case $D = D(t, X(t))$ with the difference that now the prediction time is not given explicitly in terms of P, but it is defined through an implicit relation, namely, it holds that $\sigma(t) = t + D(\sigma(t), P(t))$. Therefore, for actually computing σ one has to proceed as in the derivation of P, i.e., to differentiate relation $\sigma(\theta) = \theta + D(\sigma(\theta), P(\theta))$ and then integrate starting at the known value $\sigma(t - D(t, X(t))) = t$. It is important to note that the integral equation (4) is needed in the computation of P only when D depends on both X and t.

Backstepping Transformation and Stability Analysis

The predictor feedback designs are based on a feedback law $\kappa(X)$ that renders the closed-loop system $\dot{X} = f(X, \kappa(X))$ globally asymptotically stable. However, in the rest of the section it is assumed that the feedback law $\kappa(X)$ renders the closed-loop system $\dot{X} = f(X, \kappa(X) + v)$ input-to-state stable (ISS) with respect to v, i.e., there exists a smooth function $S : \mathbb{R}^n \to \mathbb{R}_+$ and class $\mathcal{K}_\infty$ functions $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ such that

$$\alpha_3(|X(t)|) \leq S(X(t))$$
$$\leq \alpha_4(|X(t)|) \qquad (5)$$

$$\frac{\partial S(X(t))}{\partial X} f(X(t), \kappa(X(t)))$$
$$+ v(t)) \leq -\alpha_1(|X(t)|) + \alpha_2(|v(t)|). \qquad (6)$$

Imposing this stronger assumption enables one to construct a Lyapunov functional for the

closed-loop systems (1)–(4) with the aid of the Lyapunov characterization of ISS defined in (5) and (6).

The stability analysis of the closed-loop systems (1)–(4) is explained next. Denote the infinite-dimensional backstepping transformation of the actuator state as

$$W(\theta) = U(\theta) - \kappa(P(\theta)),$$
$$\text{for all } t - D(t, X(t)) \le \theta \le t, \quad (7)$$

where $P(\theta)$ is given in terms of $U(\theta)$ from (3). Using the fact that $P(t - D(t, X(t))) = X(t)$, for all $t \ge 0$, one gets from (7) that $U(t - D(t, X(t))) = W(t - D(t, X(t))) + \kappa(X(t))$. With the fact that for all $\theta \ge 0$, $U(\theta) = \kappa(P(\theta))$ one obtains from (7) that $W(\theta) = 0$, for all $\theta \ge 0$. Yet, for all $t \le D(t, X(t))$, i.e., for all $\theta \le 0$, $W(\theta)$ might be nonzero due to the effect of the arbitrary initial condition $U(\theta)$, $\theta \in [-D(0, X(0)), 0]$. With the above observations, one can transform system (1) with the aid of transformation (7) to the following target system

$$\dot{X}(t) = f(X(t), \kappa(X(t))$$
$$+ W(t - D(t, X(t)))) \quad (8)$$
$$W(t - D(t, X(t))) = 0,$$
$$\text{for } t - D(t, X(t)) \ge 0. \quad (9)$$

Using relations (5), (6), and (8), (9) one can construct the following Lyapunov functional for showing asymptotic stability of the target system (8), (9), i.e., for the overall system consisting of the vector $X(t)$ and the transformed infinite-dimensional actuator state $W(\theta)$, $t - D(t, X(t)) \le \theta \le t$,

$$V(t) = S(X(t)) + \frac{2}{c} \int_0^{L(t)} \frac{\alpha_2(r)}{r} dr, \quad (10)$$

where $c > 0$ is arbitrary and

$$L(t) = \sup_{t - D(t, X(t)) \le \theta \le t} \left| e^{c(\sigma(\theta) - t)} W(\theta) \right|. \quad (11)$$

With the invertibility of the backstepping transformation one can then show global asymptotic stability of the closed-loop system in the original variables (X, U). In particular, there exists a class $\mathcal{KL}$ function β such that

$$|X(t)| + \sup_{t - D(t, X(t)) \le \theta \le t} |U(\theta)|$$

$$\le \beta \left(|X(0)| + \sup_{-D(0, X(0)) \le \theta \le 0} |U(\theta)|, t \right),$$

$$\text{for all } t \ge 0. \quad (12)$$

One of the main obstacles in designing globally stabilizing control laws for nonlinear systems with long input delays is the finite escape phenomenon. The input delay may be so large that the control signal cannot reach the system before its state grows unbounded. Therefore, one has to assume that the system $\dot{X} = f(X, \omega)$ is forward complete, i.e., for every initial condition and every bounded input signal the corresponding solution is defined for all $t \ge 0$.

With the forward completeness requirement, estimate (12) holds globally for constant but arbitrary large delays. For the case of time-varying delays, estimate (12) holds globally as well but under the following four conditions on the delay:

C1. $D(t) \ge 0$. This condition guarantees the causality of the system.

C2. $D(t) < \infty$. This condition guarantees that all inputs applied to the system eventually reach the system.

C3. $\dot{D}(t) < 1$. This condition guarantees that the system never feels input values that are older than the ones it has already felt, i.e., the input signal's direction never gets reversed. (This condition guarantees the existence of $\sigma = \phi^{-1}$.)

C4. $\dot{D}(t) > -\infty$ This condition guarantees that the delay cannot disappear instantaneously, but only gradually.

In the case of state-dependent delays, the delay depends on time as a result of its dependency on the state. Therefore, predictor feedback guarantees stabilization of the system when the delay satisfies the four conditions C1–C4. Yet, since

the delay is a nonnegative function of the state, conditions C2–C4 are satisfied by restricting the initial state X and the initial actuator state. Therefore estimate (12) holds locally.

Cross-References

▶ Control of Linear Systems with Delays

Recommended Reading

The main control design tool for general systems with input delays of arbitrary length is predictor feedback. The reader is referred to Artstein (1982) for the first systematic treatment of general linear systems with constant input delays. The applicability of predictor feedback was extended in Krstic (2009) to several classes of systems, such as nonlinear systems with constant input delays and linear systems with unknown input delays. Subsequently, predictor feedback was extended to general nonlinear systems with nonconstant input and state delays (Bekiaris-Liberis and Krstic 2013a). The main stability analysis tool for systems employing predictor feedback is backstepping. Backstepping was initially introduced for adaptive control of finite-dimensional nonlinear systems (Krstic et al 1995). The continuum version of backstepping was originally developed for the boundary control of several classes of PDEs in Krstic and Smyshlyaev (2008).

Bibliography

Artstein Z (1982) Linear systems with delayed controls: a reduction. IEEE Trans Autom Control 27: 869–879

Bekiaris-Liberis N, Krstic M (2013) Nonlinear control under nonconstant delays. SIAM, Philadelphia

Bekiaris-Liberis N, Krstic M (2013) Compensation of state-dependent input delay for nonlinear systems. IEEE Trans Autom Control 58: 275–289

Bekiaris-Liberis N, Krstic M (2012) Compensation of time-varying input and state delays for nonlinear systems. J Dyn Syst Meas Control 134:011009

Hale JK, Verduyn Lunel SM (1993) Introduction to functional differential equations. Springer, New York

Krstic M (2009) Delay compensation for nonlinear, adaptive, and PDE systems. Birkhauser, Boston

Krstic M, Kanellakopoulos I, Kokotovic PV (1995) Nonlinear and adaptive control design. Wiley, New York

Krstic M, Smyshlyaev A (2008) Boundary control of PDEs: a course on backstepping designs. SIAM, Philadelphia

---

Control of Quantum Systems

Ian R. Petersen
School of Engineering and Information Technology, University of New South Wales, the Australian Defence Force Academy, Canberra, Australia

Abstract

Quantum control theory is concerned with the control of systems whose dynamics are governed by the laws of quantum mechanics. Quantum control may take the form of open loop quantum control or quantum feedback control. Also, quantum feedback control may consist of measurement based feedback control, in which the controller is a classical system governed by the laws of classical physics. Alternatively, quantum feedback control may take the form of coherent feedback control in which the controller is a quantum system governed by the laws of quantum mechanics. In the area of open loop quantum control, questions of controllability along with optimal control and Lyapunov control methods are discussed. In the case of quantum feedback control, LQG and H^∞ control methods are discussed.

Keywords

Coherent quantum feedback; Measurement based quantum feedback; Quantum control; Quantum controllability

---

Introduction

Quantum control is the control of systems whose dynamics are described by the laws of quantum physics rather than classical physics. The dynamics of quantum systems must be described using quantum mechanics which allows for uniquely quantum behavior such as entanglement and coherence. There are two main approaches to quantum mechanics which are referred to as the Schrödinger picture and the Heisenberg picture. In the Schrödinger picture, quantum systems are modeled using the Schrödinger equation or a master equation which describe the evolution of the system state or density operator. In the Heisenberg picture, quantum systems are modeled using quantum stochastic differential equations which describe the evolution of system observables. These different approaches to quantum mechanics lead to different approaches to quantum control. Important areas in which quantum control problems arise include physical chemistry, atomic and molecular physics, and optics. Detailed overviews of the field o quantum control can be found in the survey papers Dong and Petersen (2010) and Brif et al. (2010) and the monographs Wiseman and Milburn (2010) and D'Alessandro (2007).

A fundamental problem in a number of approaches to quantum control is the controllability problem. Quantum controllability problems are concerned with finite dimensional quantum systems modeled using the Schrödinger picture of quantum mechanics and involves the structure of corresponding Lie groups or Lie algebras; e.g., see D'Alessandro (2007). These problems are typically concerned with closed quantum systems which are quantum systems isolated from their environment. For a controllable quantum system, an open loop control strategy can be constructed in order to manipulate the quantum state of the system in a general way. Such open loop control strategies are referred to as coherent control strategies. Time optimal control is one method of constructing these control strategies which has been applied in applications including physical chemistry and in nuclear magnetic resonance systems; e.g., see Khaneja et al. (2001).

An alternative approach to open loop quantum control is the Lyapunov approach; e.g., see Wang and Schirmer (2010). This approach extends the classical Lyapunov control approach in which a control Lyapunov function is used to construct a stabilizing state feedback control law. However in quantum control, state feedback control is not allowed since classical measurements change the quantum state of a system and the Heisenberg uncertainty principle forbids the simultaneous exact classical measurement of noncommuting quantum variables. Also, in many quantum control applications, the timescales are such that real time classical measurements are not technically feasible. Thus, in order to obtain an open loop control strategy, the deterministic closed loop system is simulated as if the state feedback control were available and this enables an open loop control strategy to be constructed. As an alternative to coherent open loop control strategies, some classical measurements may be introduced leading to incoherent control strategies; e.g., see Dong et al. (2009).

In addition to open loop quantum control approaches, a number of approaches to quantum control involve the use of feedback; e.g., see Wiseman and Milburn (2010). This quantum feedback may either involve the use of classical measurements, in which case the controller is a classical (nonquantum) system or it may involve the case where no classical measurements are used since the controller itself is a quantum system. The case in which the controller itself is a quantum system is referred to as coherent quantum feedback control; e.g., see Lloyd (2000) and James et al. (2008). Quantum feedback control may be considered using the Schrödinger picture, in which case the quantum systems under consideration are modeled using stochastic master equations. Alternatively using the Heisenberg picture, the quantum systems under consideration are modeled using quantum stochastic differential equations. Applications in which quantum feedback control can be applied include quantum optics and atomic physics. In addition, quantum control can potentially be applied to problems in quantum information (e.g., see Nielsen and Chuang 2000) such as quantum

error correction (e.g., see Kerckhoff et al. 2010) or the preparation of quantum states. Quantum information and quantum computing in turn have great potential in solving intractable computing problems such as factoring large integers using Shor's algorithm; see Shor (1994).

Schrödinger Picture Models of Quantum Systems

The state of a closed quantum system can be represented by a unit vector $|\psi\rangle$ in a complex Hilbert space $\mathcal{H}$. Such a quantum state is also referred to as a wavefunction. In the Schrödinger picture, the time evolution of the quantum state is defined by the Schrödinger equation which is in general a partial differential equation. An important class of quantum systems are finite-level systems in which the Hilbert space is finite dimensional. In this case, the Schrödinger equation is a linear ordinary differential equation of the form

$$i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle = H_0 |\psi(t)\rangle$$

where H_0 is the free Hamiltonian of the system, which is a self-adjoint operator on $\mathcal{H}$; e.g., see Merzbacher (1970). Also, $\hbar$ is the reduced Planck's constant, which can be assumed to be one with a suitable choice of units. In the case of a controlled closed quantum system, this differential equation is extended to a bilinear ordinary differential equation of the form

$$i \frac{\partial}{\partial t}|\psi(t)\rangle = \left[H_0 + \sum_{k=1}^{m} u_k(t) H_k \right] |\psi(t)\rangle \quad (1)$$

where the functions $u_k(t)$ are the control variables and the H_k are corresponding control Hamiltonians, which are also assumed to be self-adjoint operators on the underlying Hilbert space. These models are used in the open loop control of closed quantum systems.

To represent open quantum systems, it is necessary to extend the notion of quantum state to density operators ρ which are positive operators with trace one on the underlying Hilbert space

$\mathcal{H}$. In this case, the Schrödinger picture model of a quantum system is given in terms of a master equation which describes the time evolution of the density operator. In the case of an open quantum system with Markovian dynamics defined on a finite dimensional Hilbert space of dimension N, the master equation is a matrix differential equation of the form

$$\dot{\rho}(t) = -i \left[\left(H_0 + \sum_{k=1}^{m} u_k(t) H_k \right), \rho(t) \right]$$
$$+ \frac{1}{2} \sum_{j,k=0}^{N^2-1} \alpha_{j,k} \left(\left[F_j \rho(t), F_k^\dagger \right] \right.$$
$$\left. + \left[F_j, \rho(t) F_k^\dagger \right] \right);$$
$$(2)$$

e.g., see Breuer and Petruccione (2002). Here the notation $[X, \rho] = X\rho - \rho X$ refers to the commutation operator and the notation † denotes the adjoint of an operator. Also, $\{F_j\}_{j=0}^{N^2-1}$ is a basis set for the space of bounded linear operators on $\mathcal{H}$ with $F_0 = I$. Also, the matrix $A = (\alpha_{j,k})$ is assumed to be positive definite. These models, which include the Lindblad master equation for dissipative quantum systems as a special case (e.g., see Wiseman and Milburn 2010), are used in the open loop control of finite-level Markovian open quantum systems.

In quantum mechanics, classical measurements are described in terms of self-adjoint operators on the underlying Hilbert space referred to as observables; e.g., see Breuer and Petruccione (2002). An important case of measurements are projective measurements in which an observable M is decomposed as $M = \sum_{k=1}^{m} k P_k$ where the P_k are orthogonal projection operators on $\mathcal{H}$; e.g., see Nielsen and Chuang (2000). Then, for a closed quantum system with quantum state $|\psi\rangle$, the probability of an outcome k from the measurement is given by $\langle \psi | P_k | \psi \rangle$ which denotes the inner product between the vector $|\psi\rangle$ and the vector $P_k |\psi\rangle$. This notation is referred to as Dirac notation and is commonly used in quantum mechanics. If the

outcome of the quantum measurement is k, the state of the quantum system collapses to the new value of $\frac{P_k|\psi\rangle}{\sqrt{\langle\psi|P_k|\psi\rangle}}$. This change in the quantum state as a result of a measurement is an important characteristic of quantum mechanics. For an open quantum system which is in a quantum state defined by a density operator ρ, the probability of a measurement outcome k is given by $\mathrm{tr}(P_k\rho)$. In this case, the quantum state collapses to $\frac{P_k\rho P_k}{\mathrm{tr}(P_k\rho)}$.

In the case of an open quantum system with continuous measurements of an observable X, we can consider a stochastic master equation as follows:

$$
\begin{aligned}
\mathrm{d}\rho(t) = &-i\left[\left(H_0 + \sum_{k=1}^{m} u_k(t)H_k\right),\rho(t)\right]\mathrm{d}t \\
&-\kappa\left[X,[X,\rho(t)]\right]\mathrm{d}t \\
&+\sqrt{2\kappa}\left(X\rho(t)+\rho(t)X\right. \\
&\left.-2\mathrm{tr}\left(X\rho(t)\right)\rho(t)\right)\mathrm{d}W
\end{aligned}
$$

$$(3)$$

where κ is a constant parameter related to the measurement strength and $\mathrm{d}W$ is a standard Wiener increment which is related to the continuous measurement outcome $y(t)$ by

$$
\mathrm{d}W = \mathrm{d}y - 2\sqrt{\kappa}\,\mathrm{tr}\left(X\rho(t)\right)\mathrm{d}t; \qquad (4)
$$

e.g., see Wiseman and Milburn (2010). These models are used in the measurement feedback control of Markovian open quantum systems. Also, the Eqs. (3) and (4) can be regarded as a quantum filter in which $\rho(t)$ is the conditional density of the quantum system obtained by filtering the measurement signal $y(t)$; e.g., see Bouten et al. (2007) and Gough et al. (2012).

Heisenberg Picture Models of Quantum Systems

In the Heisenberg picture of quantum mechanics, the observables of a system evolve with time and the quantum state remains fixed. This picture may also be extended slightly by considering the time evolution of general operators on the underlying Hilbert space rather than just observables which are required to be self-adjoint operators. An important class of open quantum systems which are considered in the Heisenberg picture arise when the underlying Hilbert space is infinite dimensional and the system represents a collection of independent quantum harmonic oscillators interacting with a number of external quantum fields. Such linear quantum systems are described in the Heisenberg picture by linear quantum stochastic differential equations (QSDEs) of the form

$$
\begin{aligned}
\mathrm{d}x(t) &= Ax(t)\mathrm{d}t + B\mathrm{d}w(t); \\
\mathrm{d}y(t) &= Cx(t)\mathrm{d}t + D\mathrm{d}w(t) \qquad (5)
\end{aligned}
$$

where A, B, C, D are real or complex matrices, $x(t)$ is a vector of possibly noncommuting operators on the underlying Hilbert space $\mathcal{H}$; e.g., see James et al. (2008). Also, the quantity $\mathrm{d}w(t)$ is decomposed as

$$
\mathrm{d}w(t) = \beta_w(t)\mathrm{d}t + \mathrm{d}\tilde{w}(t)
$$

where $\beta_w(t)$ is an adapted process and $\tilde{w}(t)$ is a quantum Wiener process with Itô table:

$$
\mathrm{d}\tilde{w}(t)\mathrm{d}\tilde{w}(t)^{\dagger} = F_{\tilde{w}}\mathrm{d}t.
$$

Here, $F_{\tilde{w}} \geq 0$ is a real or complex matrix. The quantity $w(t)$ represents the components of the input quantum fields acting on the system. Also, the quantity $y(t)$ represents the components of interest of the corresponding output fields that result from the interaction of the harmonic oscillators with the incoming fields.

In order to represent physical quantum systems, the components of vector $x(t)$ are required to satisfy certain commutation relations of the form

$$
\left[x_j(t), x_k(t)\right] = 2i\,\Theta_{jk}, \ \ j,k = 1, 2, \ldots, n, \ \forall t
$$

where the matrix $\Theta = \left(\Theta_{jk}\right)$ is skew symmetric. The requirement to represent a physical quantum system places restrictions on the matrices A, B, C, D, which are referred to as physical

realizability conditions; e.g., see James et al. (2008) and Shaiju and Petersen (2012). QSDE models of the form (5) arise frequently in the area of quantum optics. They can also be generalized to allow for nonlinear quantum systems such as arise in the areas of nonlinear quantum optics and superconducting quantum circuits; e.g., see Bertet et al. (2012). These models are used in the feedback control of quantum systems in both the case of classical measurement feedback and in the case of coherent feedback in which the quantum controller is also a quantum system and is represented by such a QSDE model.

(S, L, H) Quantum System Models

An alternative method of modeling an open quantum system as opposed to the stochastic master equation (SME) approach or the quantum stochastic differential equation (QSDE) approach, which were considered above, is to simply model the quantum system in terms of the physical quantities which underlie the SME and QSDE models. For a general open quantum system, these quantities are the *scattering matrix S* which is a matrix of operators on the underlying Hilbert space, the coupling operator L which is a vector of operators on the underlying Hilbert space, and the system Hamiltonian which is a self-adjoint operator on the underlying Hilbert space; e.g., see Gough and James (2009). For a given (S, L, H) model, the corresponding SME model or QSDE model can be calculated using standard formulas; e.g., see Bouten et al. (2007) and James et al. (2008). Also, in certain circumstances, an (S, L, H) model can be calculated from an SME model or a QSDE model. For example, if the linear QSDE model (5) is physically realizable, then a corresponding (S, L, H) model can be found. In fact, this amounts to the definition of physical realizability.

Open Loop Control of Quantum Systems

A fundamental question in the open loop control of quantum systems is the question of controllability. For the case of a closed quantum system of the form (1), the question of controllability can be defined as follows (e.g., see Albertini and D'Alessandro 2003):

Definition 1 (Pure State Controllability) The quantum system (1) is said to be *pure state controllable* if for every pair of initial and final states $|\psi_0\rangle$ and $|\psi_f\rangle$, there exist control functions $\{u_k(t)\}$ and a time $T > 0$ such that the corresponding solution of (1) with initial condition $|\psi_0\rangle$ satisfies $|\psi(T)\rangle = |\psi_f\rangle$.

Alternative definitions have also been considered for the controllability of the quantum system (1); e.g., see Albertini and D'Alessandro (2003) and Grigoriu et al. (2013) in the case of open quantum systems. The following theorem provides a necessary and sufficient condition for pure state controllability in terms of the Lie algebra $\mathcal{L}_0$ generated by the matrices $\{-iH_0, -iH_1, \ldots, -iH_m\}$, u$(N)$ the Lie algebra corresponding to the unitary group of dimension N, su(N) the Lie algebra corresponding to the special unitary group of dimension N, sp$(\frac{N}{2})$ the $\frac{N}{2}$ dimensional symplectic group, and $\tilde{\mathcal{L}}$ the Lie algebra conjugate to sp$(\frac{N}{2})$.

Theorem 1 (See D'Alessandro 2007) *The quantum system (1) is pure state controllable if and only if the Lie algebra $\mathcal{L}_0$ satisfies one of the following conditions:*
(1) $\mathcal{L}_0 = $ su(N);
(2) $\mathcal{L}_0$ is conjugate to sp$(\frac{N}{2})$;
(3) $\mathcal{L}_0 = $ u(N);
(4) $\mathcal{L}_0 = $ span $\{i I_{N \times N}\} \oplus \tilde{\mathcal{L}}$.

Similar conditions have been obtained when alternative definitions of controllability are used.

Once it has been determined that a quantum system is controllable, the next task in open loop quantum control is to determine the control functions $\{u_k(t)\}$ which drive a given initial state to a given final state. An important approach to this problem is the optimal control approach in which a time optimal control problem is solved using Pontryagin's maximum principle to construct the control functions $\{u_k(t)\}$ which drives the given initial state to the given final state in minimum time; e.g., see Khaneja et al. (2001).

This approach works well for low dimensional quantum systems but is computationally intractable for high dimensional quantum systems.

An alternative approach for high dimensional quantum systems is the Lyapunov control approach. In this approach, a Lyapunov function is selected which provides a measure of the distance between the current quantum state and the desired terminal quantum state. An example of such a Lyapunov function is

$$V = \langle \psi(t) - \psi_f | \psi(t) - \psi_f \rangle \geq 0;$$

e.g., see Mirrahimi et al. (2005). A state feedback control law is then chosen to ensure that the time derivative of this Lyapunov function is negative. This state feedback control law is then simulated with the quantum system dynamics (1) to give the required open loop control functions $\{u_k(t)\}$.

Classical Measurement Based Quantum Feedback Control

A Schrödinger Picture Approach to Classical Measurement Based Quantum Feedback Control

In the Schrödinger picture approach to classical measurement based quantum feedback control with weak continuous measurements, we begin the stochastic master equations (3) and (4) which are considered as both a model for the system being controlled and as a filter which will form part of the final controller. These filter equations are then combined with a control law of the form

$$u(t) = f(\rho(t))$$

where the function $f(\cdot)$ is designed to achieve a particular objective such as stabilization of the quantum system. Here $u(t)$ represents the vector of control inputs $u_k(t)$. An example of such a quantum control scheme is given in the paper Mirrahimi and van Handel (2007) in which a Lyapunov method is used to design the control law $f(\cdot)$ so that a quantum system consisting of an atomic ensemble interacting with an

electromagnetic field is stabilized about a specified state $\rho_f = |\psi_m\rangle\langle\psi_m|$.

A Heisenberg Picture Approach to Classical Measurement Based Quantum Feedback Control

In this Heisenberg picture approach to classical measurement based quantum feedback control, we begin with a quantum system which is described by linear quantum stochastic equations of the form (5). In these equations, it is assumed that the components of the output vector all commute with each other and so can be regarded as classical quantities. This can be achieved if each of the components are obtained via a process of homodyne detection from the corresponding electromagnetic field; e.g., see Bachor and Ralph (2004). Also, it is assumed that the input electromagnetic field $w(t)$ can be decomposed as

$$dw(t) = \begin{bmatrix} \beta_u(t)dt + d\tilde{w}_1(t) \\ dw_2(t) \end{bmatrix} \qquad (6)$$

where $\beta_u(t)$ represents the classical control input signal and $\tilde{w}_1(t)$, $w_2(t)$ are quantum Wiener processes. The control signal displaces components of the incoming electromagnetic field acting on the system via the use of an electro-optic modulator; e.g., see Bachor and Ralph (2004).

The classical measurement feedback based controllers to be considered are classical systems described by stochastic differential equations of the form

$$dx_K(t) = A_K x_k(t)dt + B_K dy(t)$$
$$\beta_u(t)dt = C_K x_k(t)dt. \qquad (7)$$

For a given quantum system model (5), the matrices in the controller (7) can be designed using standard classical control theory techniques such as LQG control (see Doherty and Jacobs 1999) or H^∞ control (see James et al. 2008).

Coherent Quantum Feedback Control

Coherent feedback control of a quantum system corresponds to the case in which the controller itself is a quantum system which is coupled in a feedback interconnection to the quantum system being controlled; e.g., see Lloyd (2000). This type of control by interconnection is closely related to the behavioral interpretation of feedback control; e.g., see Polderman and Willems (1998).

An important approach to coherent quantum feedback control occurs in the case when the quantum system to be controlled is a linear quantum system described by the QSDEs (5). Also, it is assumed that the input field is decomposed as in (6). However in this case, the quantity $\beta_u(t)$ represents a vector of noncommuting operators on the Hilbert space underlying the controller system. These operators are described by the following linear QSDEs, which represent the quantum controller:

$$dx_K(t) = A_K x_k(t)dt + B_K dy(t) + \bar{B}_K d\bar{w}_K(t)$$
$$dy_K(t) = C_K x_k(t)dt + \bar{D}_K d\bar{w}_K(t). \qquad (8)$$

Then, the input $\beta_u(t)$ is identified as

$$\beta_u(t) = C_K x_k(t).$$

Here the quantity

$$dw_K(t) = \begin{bmatrix} dy(t) \\ d\bar{w}_K(t) \end{bmatrix} \qquad (9)$$

represents the quantum fields acting on the controller quantum system and where $w_K(t)$ corresponds to a quantum Wiener process with a given Itô table. Also, $y(t)$ represents the output quantum fields from the quantum system being controlled. Note that in the case of coherent quantum feedback control, there is no requirement that the components of $y(t)$ commute with each other and this in fact represents one of the main advantages of coherent quantum feedback control as opposed to classical measurement based quantum feedback control.

An important requirement in coherent feedback control is that the QSDEs (8) should satisfy the conditions for physical realizability; e.g., see James et al. (2008). Subject to these constraints, the controller (8) can then be designed according to an H^∞ or LQG criterion; e.g., see James et al. (2008) and Nurdin et al. (2009). In the case of coherent quantum H^∞ control, it is shown in James et al. (2008) that for any controller matrices (A_K, B_K, C_K), the matrices $(\bar{B}_K, \bar{D}_K)$ can be chosen so that the controller QSDEs (8) are physically realizable. Furthermore, the choice of the matrices $(\bar{B}_K, \bar{D}_K)$ does not affect the H^∞ performance criterion considered in James et al. (2008). This means that the coherent controller can be designed using the same approach as designing a classical H^∞ controller.

In the case of coherent LQG control such as considered in Nurdin et al. (2009), the choice of the matrices $(\bar{B}_K, \bar{D}_K)$ significantly affects the closed loop LQG performance of the quantum control system. This means that the approach used in solving the coherent quantum H^∞ problem given in James et al. (2008) cannot be applied to the coherent quantum LQG problem. To date there exist only some nonconvex optimization methods which have been applied to the coherent quantum LQG problem (e.g., see Nurdin et al. 2009), and the general solution to the coherent quantum LQG control problem remains an open question.

Cross-References

▶ Bilinear Control of Schrödinger PDEs
▶ Robustness Issues in Quantum Control

Bibliography

Albertini F, D'Alessandro D (2003) Notions of controllability for bilinear multilevel quantum systems. IEEE Trans Autom Control 48:1399–1403

Bachor H, Ralph T (2004) A guide to experiments in quantum optics, 2nd edn. Wiley-VCH, Weinheim

Bertet P, Ong FR, Boissonneault M, Bolduc A, Mallet F, Doherty AC, Blais A, Vion D, Esteve D (2012) Circuit quantum electrodynamics with a nonlinear resonator.

In: Dykman M (ed) Fluctuating nonlinear oscillators: from nanomechanics to quantum superconducting circuits. Oxford University Press, Oxford

Breuer H, Petruccione F (2002) The theory of open quantum systems. Oxford University Press, Oxford

Brif C, Chakrabarti R, Rabitz H (2010) Control of quantum phenomena: past, present and future. New J Phys 12:075008

Bouten L, van Handel R, James M (2007) An introduction to quantum filtering. SIAM J Control Optim 46(6):2199–2241

D'Alessandro D (2007) Introduction to quantum control and dynamics. Chapman & Hall/CRC, Boca Raton

Doherty A, Jacobs K (1999) Feedback-control of quantum systems using continuous state-estimation. Phys Rev A 60:2700–2711

Dong D, Petersen IR (2010) Quantum control theory and applications: a survey. IET Control Theory Appl 4(12):2651–2671

Dong D, Lam J, Tarn T (2009) Rapid incoherent control of quantum systems based on continuous measurements and reference model. IET Control Theory Appl 3:161–169

Gough J, James MR (2009) The series product and its application to quantum feedforward and feedback networks. IEEE Trans Autom Control 54(11):2530–2544

Gough JE, James MR, Nurdin HI, Combes J (2012) Quantum filtering for systems driven by fields in single-photon states or superposition of coherent states. Phys Rev A 86:043819

Grigoriu A, Rabitz H, Turinici G (2013) Controllability analysis of quantum systems immersed within an engineered environment. J Math Chem 51(6):1548–1560

James MR, Nurdin HI, Petersen IR (2008) H^∞ control of linear quantum stochastic systems. IEEE Trans Autom Control 53(8):1787–1803

Kerckhoff J, Nurdin HI, Pavlichin DS, Mabuchi H (2010) Designing quantum memories with embedded control: photonic circuits for autonomous quantum error correction. Phys Rev Lett 105:040502

Khaneja N, Brockett R, Glaser S (2001) Time optimal control in spin systems. Phys Rev A 63:032308

Lloyd S (2000) Coherent quantum feedback. Phys Rev A 62:022108

Merzbacher E (1970) Quantum mechanics, 2nd edn. Wiley, New York

Mirrahimi M, van Handel R (2007) Stabilizing feedback controls for quantum systems. SIAM J Control Optim 46(2):445–467

Mirrahimi M, Rouchon P, Turinici G (2005) Lyapunov control of bilinear Schrödinger equations. Automatica 41:1987–1994

Nielsen M, Chuang I (2000) Quantum computation and quantum information. Cambridge University Press, Cambridge, UK

Nurdin HI, James MR, Petersen IR (2009) Coherent quantum LQG control. Automatica 45(8):1837–1846

Polderman JW, Willems JC (1998) Introduction to mathematical systems theory: a behavioral approach. Springer, New York

Shaiju AJ, Petersen IR (2012) A frequency domain condition for the physical realizability of linear quantum systems. IEEE Trans Autom Control 57(8):2033–2044

Shor P (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Goldwasser S (ed) Proceedings of the 35th annual symposium on the foundations of computer science. IEEE Computer Society, Los Alamitos, pp 124–134

Wang W, Schirmer SG (2010) Analysis of Lyapunov method for control of quantum states. IEEE Trans Autom Control 55(10):2259–2270

Wiseman HM, Milburn GJ (2010) Quantum measurement and control. Cambridge University Press, Cambridge, UK

Control of Ship Roll Motion

Tristan Perez[1] and Mogens Blanke[2,3]
[1]Electrical Engineering & Computer Science, Queensland University of Technology, Brisbane, QLD, Australia
[2]Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark (DTU), Lyngby, Denmark
[3]Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, Trondheim, Norway

Abstract

The undesirable effects of roll motion of ships (rocking about the longitudinal axis) became noticeable in the mid-nineteenth century when significant changes were introduced to the design of ships as a result of sails being replaced by steam engines and the arrangement being changed from broad to narrow hulls. The combination of these changes led to lower transverse stability (lower restoring moment for a given angle of roll) with the consequence of larger roll motion. The increase in roll motion and its effect on cargo and human performance lead to the development several control devices that aimed at reducing and controlling roll motion. The control devices most commonly used today are fin stabilizers, rudder, anti-roll tanks, and gyrostabilizers. The use of

different types of actuators for control of ship roll motion has been amply demonstrated for over 100 years. Performance, however, can still fall short of expectations because of difficulties associated with control system design, which have proven to be far from trivial due to fundamental performance limitations and large variations of the spectral characteristics of wave-induced roll motion. This short article provides an overview of the fundamentals of control design for ship roll motion reduction. The overview is limited to the most common control devices. Most of the material is based on Perez (Ship motion control. Advances in industrial control. Springer, London, 2005) and Perez and Blanke (Ann Rev Control 36(1):1367–5788, 2012).

Keywords

Roll damping; Ship motion control

Ship Roll Motion Control Techniques

One of the most commonly used devices to attenuate ship motion are the fin stabilisers. These are small controllable fins located on the bilge of the hull usually amid ships. These devices attain a performance in the range of 60–90 % of roll reduction (root mean square) (Sellars and Martin 1992). They require control systems that sense the vessel's roll motion and act by changing the angle of the fins. These devices are expensive and introduce underwater noise that can affect sonar performance, they add to propulsion losses, and they can be damaged. Despite this, they are among the most commonly used ship roll motion control device. From a control perspective, highly nonlinear effects (dynamic stall) may appear when operating in severe sea states and heavy rolling conditions (Gaillarde 2002).

During studies of ship damage stability conducted in the late 1800s, it was observed that under certain conditions the water inside the vessel moved out of phase with respect to the wave profile, and thus, the weight of the water on the vessel counteracted the increase of pressure on the hull, hence reducing the net roll excitation moment. This led to the development of fluid anti-roll tank stabilizers. The most common type of anti-roll tank is the U-tank, which comprises two reservoirs, located one on port and one on starboard, connected at the bottom by a duct. Anti-roll tanks can be either passive or active. In passive tanks, the fluid flows freely from side to side. According to the density and viscosity of the fluid used, the tank is dimensioned so that the time required for most of the fluid to flow from side to side equals the natural roll period of the ship. Active tanks operate in a similar manner, but they incorporate a control system that modifies the natural period of the tank to match the actual ship roll period. This is normally achieved by controlling the flow of air from the top of one reservoir to the other. Anti-roll tanks attain a medium to high performance in the range of 20–70 % of roll angle reduction (RMS) (Marzouk and Nayfeh 2009). Anti-roll tanks increase the ship displacement. They can also be used to correct list (steady-state roll angle), and they are the preferred stabilizer for icebreakers.

Rudder-roll stabilization (RRS) is a technique based on the fact that the rudder is located not only aft, but also below the center of gravity of the vessel, and thus the rudder imparts not only yaw but also roll moment. The idea of using the rudder for simultaneous course keeping and roll reduction was conceived in the late 1960s by observations of anomalous behavior of autopilots that did not have appropriate wave filtering – a feature of the autopilot that prevents the rudder from reacting to every single wave; see, for example, Fossen and Perez (2009) for a discussion on wave filtering. Rudder-roll stabilization has been demonstrated to attain medium to high performance in the range of 50–75 % of roll reduction (RMS) (Baitis et al. 1983; Blanke et al. 1989; Källström et al. 1988; Oda et al. 1992; van Amerongen et al. 1990). The upgrade of the rudder machinery is required to be able to attain slew rates in the range 10–20 deg/s for RRS to have sufficient control authority.

A gyrostabilizer uses the gyroscopic effects of large rotating wheels to generate a roll reducing torque. The use of gyroscopic effects was

proposed in the early 1900s as a method to eliminate roll, rather than to reduce it. Although the performance of these systems was remarkable, up to 95 % roll reduction, their high cost, the increase in weight, and the large stress produced on the hull masked their benefits and prevented further developments. However, a recent increase in development of gyrostabilizers has been seen in the yacht industry (Perez and Steinmann 2009).

Fins and rudder give rise to lift forces in proportion to the square of flow velocity past the fin. Hence, roll stabilization by fin or rudder is not possible at low or zero speed. Only U-tanks and gyro devices are able to provide stabilization in these conditions. For further details about the performance of different devices, see Sellars and Martin (1992), and for a comprehensive description of the early development of devices, see Chalmers (1931).

Modeling of Ship Roll Motion for Control Design

The study of roll motion dynamics for control system design is normally done in terms of either one- or four-degrees-of-freedom (DOF) models. The choice between models of different complexity depends on the type of motion control system considered.

For a one-degree-of-freedom (1DOF) case, the following model is used:

$$\dot{\phi} = p, \tag{1}$$

$$I_{xx} \dot{p} = K_h + K_w + K_c, \tag{2}$$

where ϕ is roll angle, p is roll rate, and I_{xx} is rigid-body moment of inertia about the x-axis of a body-fixed coordinate system, where K_h is hydrostatic and hydrodynamic torques, K_w torque generated by wave forces acting on the hull, and K_c the control torques. The hydrodynamic torque can be approximated by the following parametric model: $K_h \approx K_{\dot{p}} \dot{p} + K_p p + K_{p|p|} p|p| + K(\phi)$. The first term represents a hydrodynamic torque in roll due to pressure change that is proportional to the roll accelerations, and the coefficient $K_{\dot{p}}$

is called roll added mass (inertia). The second term is a damping term, which captures forces due to wave making and linear skin friction, and the coefficient K_p is a linear damping coefficient. The third term is a nonlinear damping term, which captures forces due to viscous effects. The last term is the restoring torque due to gravity and buoyancy.

For a 4DOF model (surge, sway, roll, and yaw), motion variables considered are $\boldsymbol{\eta} = [\phi \ \psi]^\mathsf{T}$, $\boldsymbol{\nu} = [u \ v \ p \ r]^\mathsf{T}$, $\boldsymbol{\tau}_i = [X \ Y \ K \ N]^\mathsf{T}$, where ψ is the yaw angle, the body-fixed velocities are u-surge and v-sway, and r is the yaw rate. The forces and torques are X-surge, Y-sway, K-roll, and N-yaw. With these variables, the following mathematical model is usually considered:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta}) \, \boldsymbol{\nu}, \tag{3}$$

$$\mathbf{M}_{RB} \, \dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_h + \boldsymbol{\tau}_c + \boldsymbol{\tau}_d, \tag{4}$$

where $\mathbf{J}(\boldsymbol{\eta})$ is a kinematic transformation, $\mathbf{M}_{RB}$ is the rigid-body inertia matrix that corresponds to expressing the inertia tensor in body-fixed coordinates, $\mathbf{C}_{RB}(\boldsymbol{\nu})$ is the rigid-body Coriolis and centripetal matrix, and $\boldsymbol{\tau}_h$, $\boldsymbol{\tau}_c$, and $\boldsymbol{\tau}_d$ represent the hydrodynamic, control, and disturbance vector of force components and torques, respectively.

The hydrostatic and hydrodynamic forces are $\boldsymbol{\tau}_h \approx -\mathbf{M}_A \, \dot{\boldsymbol{\nu}} - \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{K}(\phi)$. The first two terms have origin in the motion of a vessel in an irrotational flow in a nonviscous fluid. The third term corresponds to damping forces due to potential (wave making), skin friction, vortex shedding, and circulation (lift and drag). The hydrodynamic effects involved are quite complex, and different approaches based on superposition of either odd-term Taylor expansions or square modulus ($x|x|$) series expansions are usually considered Abkowitz (1964) and Fedyaevsky and Sobolev (1964). The $\mathbf{K}(\phi)$ term represents the restoring forces in roll due to buoyancy and gravity. The 4DOF model captures parameter dependency on ship speed as well as the couplings between steering and roll, and it is useful for controller design. For additional details about mathematical model of marine vehicles, see Fossen (2011).

Wave-Disturbance Models

The action of the waves creates changes in pressure on the hull of the ship, which translate into forces and moments. It is common to model the ship motion response due to waves within a linear framework and to obtain two frequency-response functions (FRF), wave to excitation $F_i(j\omega, U, \chi)$ and wave to motion $H_i(j\omega, U, \chi)$ response functions, where i indicates the degree of freedom. These FRF depend on the wave frequency, the ship speed, and the angle χ at which the waves encounter the ship – this is called the encounter angle.

The wave elevation in deep water is approximately a stochastic process that is zero mean, stationary for short periods of time, and Gaussian (Haverre and Moan 1985). Under these assumptions, the wave elevation ζ is fully described by a power spectral density $\Phi_{\zeta\zeta}(\omega)$. With a linear response assumption, the power spectral density of wave to excitation force and wave to motion can be expressed as

$$\Phi_{FF,i}(j\omega) = |F_i(j\omega, U, \chi)|^2 \Phi_{\zeta\zeta}(j\omega),$$

$$\Phi_{\eta\eta,i}(j\omega) = |H_i(j\omega, U, \chi)|^2 \Phi_{\zeta\zeta}(j\omega).$$

These spectra are models of the wave-induced forces and motions, respectively, from which it its common to generate either time series of wave excitation forces in terms of the encounter frequency to be used as input disturbances in simulation models or time series of wave-induced motion to be used as output disturbance; see, for example, Perez (2005) and references herein.

Roll Motion Control and Performance Limitations

The analysis of performance of ship roll motion control by means of force actuators is usually conducted within a linear framework by linearizing the models. For a SISO loop where the wave-induced roll motion is considered an output disturbance, the Bode integral constraint applies. This imposes restrictions on one's freedom to shape the closed-loop transfer function

to attenuate the motion due to the wave-induced forces in different frequency ranges. These results have important consequences on the design of a roll motion control system since the frequency of the waves seen from the vessel changes significantly with the sea state, the speed of the vessel, and the wave encounter angle. The changing characteristics on open-loop roll motion in conjunction with the Bode integral constraint make the control design challenging since roll amplification may occur if the control design is not done properly. For some roll motion control problems, like using the rudder for simultaneous roll attenuation and heading control, the system presents non-minimum phase dynamics. In this case, the trade-off of reduced sensitivity *vs.* amplification of roll motion is dominating at frequencies close to the non-minimum phase zero – a constraint with origin in the Poisson integral (Hearns and Blanke 1998); see also Perez (2005).

It should be noted that non-minimum phase dynamics also occurs with fin stabilizers, when the stabilizers are located aft of the center of gravity. With the fins at this location, they behave like a rudder and introduce non-minimum phase dynamics and heading interference at low wave-excitation frequencies. These aspects of fin location were discussed by Lloyd (1989).

The above discussion highlights general design constraints that apply to roll motion control systems in terms of the dynamics of the vessel and actuator. In addition to these constraints, one needs also to account for limitations in actuator slew rate and angle.

Controls Techniques Used in Different Roll Control Systems

Fin Stabilizers

In regard to fin stabilizers, the control design is commonly address using the 1DOF model (1) and (2). The main issues associated with control design are the parametric uncertainty in model and the Bode integral constraint. This integral constraint can lead to roll amplification due to changes in the spectrum of the wave-induced

roll moment with sea state and sailing conditions (speed and encounter angle). Fin machinery is designed so that the rate of the fin motion is fast enough, and actuator rate saturation is not an issue in moderate sea states. The fins could be used to correct heeling angles (steady-state roll) when the ship makes speed, but this is avoided due to added resistance. If it is used, integral action needs to include anti-windup. In terms of control strategies, PID, $\mathcal{H}_\infty$, and LQR techniques have been successfully applied in practice. Highly nonlinear effects (dynamic stall) may appear when operating in severe sea states and heavy rolling conditions, and proposals for applications of model predictive control have been put forward to constraint the effective angle of attack of the fins. In addition, if the fins are located too far aft along the ship, the dynamic response from fin angle to roll can exhibit non-minimum phase dynamics, which can limit the performance at low encounter frequencies. A thorough review of the control literature can be found in Perez and Blanke (2012).

Rudder-Roll Stabilization

The problem of rudder-roll stabilization requires the 4DOF model (3) and (4), which captures the interaction between roll, sway, and yaw together with the changes in the hydrodynamic forces due to the forward speed. The response from rudder to roll is non-minimum phase (NMP), and the system is characterized by further constraints due to the single-input-two-output nature of the control problem – attenuate roll without too much interference with the heading. Studies of fundamental limitations due to NMP dynamics have been approached using standard frequency-domain tools by Hearns and Blanke (1998) and Perez (2005). A characterization of the trade-off between roll reduction vs. increase of interference was part of the controller design in Stoustrup et al. (1994). Perez (2005) determined the limits obtainable using optimal control with full disturbance information. The latter also incorporated constraints due to the limiting authority of the control action in rate and magnitude of rudder machinery and stall conditions of the rudder. The control design for rudder-roll stabilization

has been addressed in practice using PID, LQG, and $\mathcal{H}_\infty$ and standard frequency-domain linear control designs. The characteristics of limited control authority were solved by van Amerongen et al. (1990) using automatic gain control. In the literature, there have been proposals put forward for the use of model predictive control, QFT, sliding-mode nonlinear control, and auto-regressive stochastic control. Combined use of fin and rudder has also be investigated. Grimble et al. (1993) and later Roberts et al. (1997) used $\mathcal{H}_\infty$ control techniques. Thorough comparison of controller performances for warships was published in Crossland (2003). A thorough review of the control literature can be found in Perez and Blanke (2012).

Gyrostabilizers

Using a single gimbal suspension gyrostabilizer for roll damping control, the coupled vessel-roll-gyro model can be modeled as follows:

$$\dot{\phi} = p, \tag{5}$$

$$K_{\dot{p}}\,\dot{p} + K_p\,p + K_\phi\,\phi = K_w - K_g\dot{\alpha}\cos\alpha \tag{6}$$

$$I_p\ddot{\alpha} + B_p\dot{\alpha} + C_p\sin\alpha = K_g\,p\cos\alpha + T_p, \tag{7}$$

where (6) represents the 1DOF roll dynamics and (7) represents the dynamics of the gyrostabilizer about the axis of the gimbal suspension, where α is the gimbal angle, equivalent to the precession angle for a single gimbal suspension, I_p is gimbal and wheel inertia about the gimbal axis, B_p is the damping, and C_p is a restoring term of the gyro about the precession axis due to location of the gyro center of mass relative to the precession axis (Arnold and Maunder 1961). T_p is the control torque applied to the gimbal. The use of twin counter-spinning wheels prevents gyroscopic coupling with other degrees of freedom. Hence, the control design for gyrostabilizers can be based on a linear single-degree-of-freedom model for roll.

The wave-induced roll moment K_w excites the roll motion. As the roll motion develops, the roll rate p induces a torque along the precession axis of the gyrostabilizer. As the precession angle α

develops, there is reaction torque done on the vessel that opposes the wave-induced moment. The later is the roll stabilizing torque, $X_g \triangleq -K_g \dot{\alpha} \cos \alpha \approx -K_g \dot{\alpha}$. This roll torque can only be controlled indirectly through the precession dynamics in (7) via T_p. In the model above, the spin angular velocity ω_{spin} is controlled to be constant; hence the wheels' angular momentum $K_g = I_{spin} \omega_{spin}$ is constant.

The precession control torque T_p is used to control the gyro. As observed by Sperry (Chalmers 1931), the intrinsic behavior of the gyrostabilizer is to use roll rate to generate a roll torque. Hence, one could design a precession torque controller such that from the point of view of the vessel, the gyro behaves as damper. Depending on how precession torque is delivered, it may be necessary to constraint precession angle and rate. This problem has been recently considered in Donaire and Perez (2013) using passivity-based control.

U-tanks

U-tanks can be passive or active. Roll reduction is achieved by attempting to transfer energy from the roll motion to motion of liquid within the tank and using the weight of the liquid to counteract the wave excitation moment. A key aspect of the design is the dimension and geometry of the tank to ensure that there is enough weight due to the displaced liquid in the tank and that the oscillation of the fluid in the tank matches the vessel natural frequency in roll; see Holden and Fossen (2012) and references herein. The design of the U-tank can ensure a single-frequency matching, at which the performance is optimized, and for this frequency the roll natural frequency is used. As the frequency of roll motion departs from this, a degradation of roll reduction occurs. Active U-tanks use valves to control the flow of air from the top of the reservoirs to extend the frequency matching in sailing conditions in which the roll dominant frequency is lower than the roll natural frequency – the flow of air is used to delay the motion of the liquid from one reservoir to the other. This control is achieved by detecting the dominant roll frequency and using this information to control the air flow from one reservoir

to the other. If the roll dominant frequency is higher than the roll natural frequency, the U-tank is used in passive mode, and the standard roll reduction degradation occurs.

Summary and Future Directions

This article provides a brief summary of control aspects for the most common ship roll motion control devices. These aspects include the type of mathematical models used to design and analyze the control problem, the inherent fundamental limitations and the constraints that some of the designs are subjected to, and the performance that can be expected from the different devices. As an outlook, one of the key issues in roll motion control is the model uncertainty and the adaptation to the changes in the environmental conditions. As the vessel changes speed and heading, or as the seas build up or abate, the dominant frequency range of the wave-induced forces changes significantly. Due to the fundamental limitations discussed, a nonadaptive controller may produce roll amplification rather than roll reduction. This topic has received some attention in the literature via multi-mode control switching, but further work in this area could be beneficial. In the recent years, new devices have appeared for stabilization at zero speed, like flapping fins and rotating cylinders. Also the industry's interest in roll gyrostabilizers has been re-ignited. The investigation of control designs for these devices has not yet received much attention within the control community. Hence, it is expected that this will create a potential for research activity in the future.

Cross-References

▶ Fundamental Limitation of Feedback Control
▶ H-Infinity Control
▶ H_2 Optimal Control
▶ Linear Quadratic Optimal Control
▶ Mathematical Models of Ships and Underwater Vehicles

Bibliography

Abkowitz M (1964) Lecture notes on ship hydrodynamics–steering and manoeuvrability. Technical report Hy-5, Hydro and Aerodynamics Laboratory, Lyngby

Arnold R, Maunder L (1961) Gyrodynamics and its engineering applications. Academic, New York/London

Baitis E, Woollaver D, Beck T (1983) Rudder roll stabilization of coast guard cutters and frigates. Nav Eng J 95(3):267–282

Blanke M, Haals P, Andreasen KK (1989) Rudder roll damping experience in Denmark. In: Proceedings of IFAC workshop CAMS'89, Lyngby

Chalmers T (1931) The automatic stabilisation of ships. Chapman and Hall, London

Crossland P (2003) The effect of roll stabilization controllers on warship operational performance. Control Eng Pract 11:423–431

Donaire A, Perez T (2013) Energy-based nonlinear control of ship roll gyro-stabiliser with precession angle constraints. In: 9th IFAC conference on control applications in marine systems, Osaka

Fedyaevsky K, Sobolev G (1964) Control and stability in ship design. State Union Shipbuilding, Leningrad

Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

Fossen T, Perez T (2009) Kalman filtering for positioning and heading control of ships and offshore rigs. IEEE Control Syst Mag 29(6):32–46

Gaillarde G (2002) Dynamic behavior and operation limits of stabilizer fins. In: IMAM international maritime association of the Mediterranean, Creta

Grimble M, Katebi M, Zang Y (1993) $\mathcal{H}_\infty$–based ship fin-rudder roll stabilisation. In: 10th ship control system symposium SCSS, Ottawa, vol 5, pp 251–265

Haverre S, Moan T (1985) On some uncertainties related to short term stochastic modelling of ocean waves. In: Probabilistic offshore mechanics. Progress in engineering science. CML

Hearns G, Blanke M (1998) Quantitative analysis and design of rudder roll damping controllers. In: Proceedings of CAMS'98, Fukuoka, pp 115–120

Holden C, Fossen TI (2012) A nonlinear 7-DOF model for U-tanks of arbitrary shape. Ocean Eng 45: 22–37

Källström C, Wessel P, Sjölander S (1988) Roll reduction by rudder control. In: Spring meeting-STAR symposium, 3rd IMSDC, Pittsburgh

Lloyd A (1989) Seakeeping: ship behaviour in rough weather. Ellis Horwood

Marzouk OA, Nayfeh AH (2009) Control of ship roll using passive and active anti-roll tanks. Ocean Eng 36:661–671

Oda H, Sasaki M, Seki Y, Hotta T (1992) Rudder roll stabilisation control system through multivariable autoregressive model. In: Proceedings of IFAC conference on control applications of marine systems–CAMS

Perez T (2005) Ship motion control. Advances in industrial control. Springer, London

Perez T, Blanke M (2012) Ship roll damping control. Ann Rev Control 36(1):1367–5788

Perez T, Steinmann P (2009) Analysis of ship roll gyrostabiliser control. In: 8th IFAC international conference on manoeuvring and control of marine craft, Guaruja

Roberts G, Sharif M, Sutton R, Agarwal A (1997) Robust control methodology applied to the design of a combined steering/stabiliser system for warships. IEE Proc Control Theory Appl 144(2):128–136

Sellars F, Martin J (1992) Selection and evaluation of ship roll stabilization systems. Mar Technol SNAME 29(2):84–101

Stoustrup J, Niemann HH, Blanke M (1994) Rudder-roll damping for ships- a new $\mathcal{H}_\infty$ approach. In: Proceedings of 3rd IEEE conference on control applications, Glasgow, pp 839–844

van Amerongen J, van der Klugt P, van Nauta Lemke H (1990) Rudder roll stabilization for ships. Automatica 26:679–690

Control Structure Selection

Sigurd Skogestad
Department of Chemical Engineering,
Norwegian University of Science and
Technology (NTNU), Trondheim, Norway

Abstract

Control structure selection deals with selecting what to control (outputs), what to measure and what to manipulate (inputs), and also how to split the controller in a hierarchical and decentralized manner. The most important issue is probably the selection of the controlled variables (outputs), CV = Hy, where y are the available measurements and H is a degree of freedom that is seldom treated in a systematic manner by control engineers. This entry discusses how to find H for both for the upper (slower) economic layer and the lower (faster) regulatory layer in the control hierarchy. Each layer may be split in a decentralized fashion. Systematic approaches for input/output (IO) selection are presented.

Keywords

Control configuration; Control hierarchy; Control structure design; Decentralized control;

Economic control; Input-output controllability; Input/output selection; Plantwide control; Regulatory control; Supervisory control

Introduction

Consider the generalized controller design problem in Fig. 1 where P denotes the generalized plant model. Here, the objective is to design the controller K, which, based on the sensed outputs v, computes the inputs (MVs) u such that the variables z are kept small, in spite of variations in the variables w, which include disturbances (d), varying setpoints/references (CV_s) and measurement noise (n),

$$w = [d, CV_s, n]$$

The variables z, which should be kept small, typically include the control error for the selected controlled variables (CV) plus the plant inputs (u),

$$z = [CV - CV_s; u]$$

The variables v, which are the inputs to the controller, include all known variables, including measured outputs (y_m), measured disturbances (d_m) and setpoints,

$$v = [y_m; d_m; CV_s].$$

The cost function for designing the optimal controller K is usually the weighted control error,
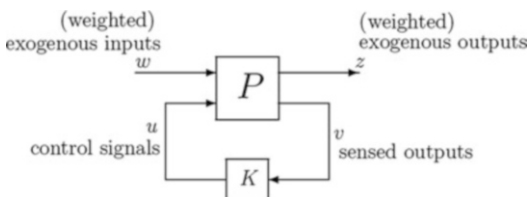
Control Structure Selection, Fig. 1 General formulation for designing the controller K. The plant P is controlled by manipulating u, and is disturbed by the signals w. The controller uses the measurements v, and the control objective is to keep the outputs (weighted control error) z as small as possible

$J' = ||W'z||$. The reason for using a prime on J (J'), is to distinguish it from the economic cost J which we later use for selecting the controlled variables (CV).

Notice that it is assumed in Fig. 1 that we know what to measure (v), manipulate (u), and, most importantly, which variables in z we would like to keep at setpoints (CV), that is, we have assumed a given control structure. The term "control structure selection" (CSS) and its synonym "control structure design" (CSD) is associated with the overall *control philosophy* for the system with emphasis on the **structural decisions** which are a prerequisite for the controller design problem in Fig. 1:

1. *Selection of controlled variables (CVs, "outputs," included in z in Fig. 1)*
2. *Selection of manipulated variables (MVs, "inputs," u in Fig. 1)*
3. *Selection of measurements y (included in v in Fig. 1)*
4. *Selection of control **configuration** (structure of overall controller K that interconnects the controlled, manipulated and measured variables; structure of K in Fig. 1)*
5. *Selection of type of controller K (PID, MPC, LQG, H-infinity, etc.) and objective function (norm) used to design and analyze it.*

Decisions 2 and 3 (selection of u and y) are sometimes referred to as the input/output (IO) selection problem. In practice, the controller (K) is usually divided into several layers, operating on different time scales (see Fig. 2), which implies that we in addition to selecting the (primary) controlled variables ($CV_1 \equiv CV$) must also select the (secondary) variables that interconnect the layers (CV_2).

Control structure selection includes all the *structural* decisions that the engineer needs to make when designing a control system, but it does not involve the actual design of each individual controller block. Thus, it involves the decisions necessary to make a block diagram (Fig. 1; used by control engineers) or process & instrumentation diagram (used by process engineers) for the entire plant, and provides the starting point for a detailed controller design.
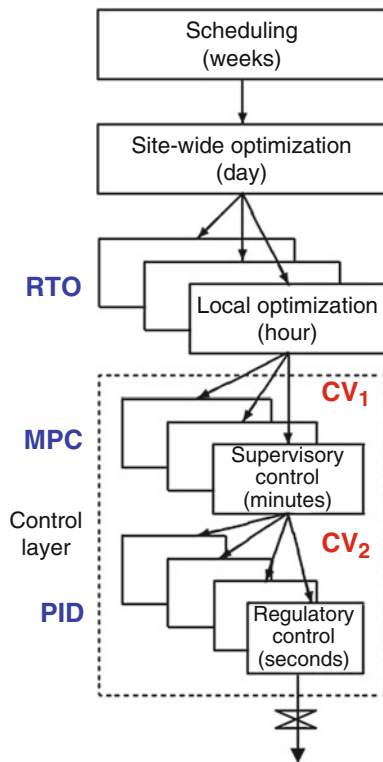
Control Structure Selection, Fig. 2 Typical control hierarchy, as illustrated for a process plant

The term "plantwide control," which is a synonym for "control structure selection," is used in the field of process control. Control structure selection is particularly important for process control because of the complexity of large processing plants, but it applies to all control applications, including vehicle control, aircraft control, robotics, power systems, biological systems, social systems, and so on.

It may be argued that control structure selection is more important than the controller design itself. Yet, control structure selection is hardly covered in most control courses. This is probably related to the complexity of the problem, which requires the knowledge from several engineering fields. In the mathematical sense, the control structure selection problem is a formidable combinatorial problem which involves a large number of discrete decision variables.

Overall Objectives for Control and Structure of the Control Layer

The starting point for control system design is to define clearly the operational objectives. There are usually two main objectives for control:

1. Longer-term economic operation (minimize economic cost J subject to satisfying operational constraints)
2. Stability and short-term regulatory control

The first objective is related to "making the system operate as intended," where economics are an important issue. Traditionally, control engineers have not been much involved in this step. The second objective is related to "making sure the system stays operational," where stability and robustness are important issues, and this has traditionally been the main domain of control engineers. In terms of designing the control system, the second objective (stabilization) is usually considered first. An example is bicycle riding; we first need to learn how to stabilize the bicycle (regulation), before trying to use it for something useful (optimal operation), like riding to work and selecting the shortest path.

We use the term "economic cost," because usually the cost function J can be given a monetary value, but more generally, the cost J could be any scalar cost. For example, the cost J could be the "environmental impact" and the economics could then be given as constraints.

In theory, the optimal strategy is to combine the control tasks of optimal economic operation and stabilization/regulation in a single centralized controller K, which at each time step collects all the information and computes the optimal input changes. In practice, simpler controllers are used. The main reason for this is that in most cases one can obtain acceptable control performance with simple structures, where each controller block involves only a few variables. Such control systems can be designed and tuned with much less effort, especially when it comes to the modeling and tuning effort.

So how are large-scale systems controlled in practise? Usually, the controller K is decomposed

into several subcontrollers, using two main principles

– *Decentralized (local) control.* This "horizontal decomposition" of the control layer is usually based on separation in space, for example, by using local control of individual units.
– *Hierarchical (cascade) control.* This "vertical decomposition" is usually based on time scale separation, as illustrated for a process plant in Fig. 2. The upper three layers in Fig. 2 deal explicitly with economic optimization and are not considered here. We are concerned with the two lower *control layers,* where the main objective is to track the setpoints specified by the layer above.

In accordance with the two main objectives for control, the control layer is in most cases divided hierarchically in two layers (Fig. 2):

1. A "slow" supervisory (economic) layer
2. A "fast" regulatory (stabilization) layer

Another reason for the separation in two control layers, is that the tasks of economic operation and regulation are fundamentally different. Combining the two objectives in a single cost function, which is required for designing a single centralized controller K, is like trying to compare apples and oranges. For example, how much is an increased stability margin worth in monitory units [$]? Only if there is a reasonable benefit in combining the two layers, for example, because there is limited time scale separation between the tasks of regulation and optimal economics, should one consider combining them into a single controller.

Notation and Matrices H_1 and H_2 for Controlled Variable Selection

The most important notation is summarized in Table 1 and Fig. 3. To distinguish between the two control layers, we use "1" for the upper supervisory (economic) layer and "2" for the regulatory layer, which is "secondary" in terms of its place in the control hierarchy.

There is often limited possibility to select the input set (u) as it is usually constrained by the
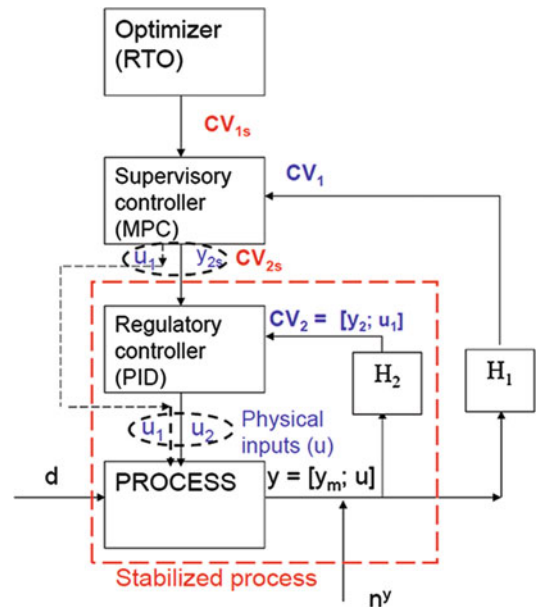
Control Structure Selection, Fig. 3 Block diagram of a typical control hierarchy, emphasizing the selection of controlled variables for supervisory (economic) control ($CV_1 = H_1 y$) and regulatory control ($CV_2 = H_2 y$)

Control Structure Selection, Table 1 Important notation

$u = [u_1; u_2]$ = set of all available physical plant inputs
u_1 = inputs used directly by supervisory control layer
u_2 = inputs used by regulatory layer
y_m = set of all measured outputs
$y = [y_m; u]$ = combined set of measurements and inputs
y_2 = controlled outputs in regulatory layer (subset or combination of y); $\dim(y_2) = \dim(u_2)$
$CV_1 = H_1 y$ = controlled variables in supervisory layer; $\dim(CV_1) = \dim(u)$
$CV_2 = [y_2; u_1] = H_2 y$ = controlled variables in regulatory layer; $\dim(CV_2) = \dim(u)$
$MV_1 = CV_{2s} = [y_{2s}; u_1]$ = manipulated variables in supervisory layer; $\dim(MV_1) = \dim(u)$
$MV_2 = u_2$ = manipulated variables in regulatory layer; $\dim(MV_2) = \dim(u_2) \leq \dim(u)$

plant design. However, there may be a possibility to add inputs or to move some to another location, for example, to avoid saturation or to reduce the time delay and thus improve the input-output controllability.

There is much more flexibility in terms of output selection, and the most important structural decision is related to the selection of controlled variables in the two control layers, as given by the decision matrices H_1 and H_2 (see Fig. 3).

$$CV_1 = H_1 y$$
$$CV_2 = H_2 y$$

Note from the definition in Table 1 that $y = [y_m; u]$. Thus, y includes, in addition to the candidate measured outputs (y_m), also the physical inputs u. This allows for the possibility of selecting an input u as a "controlled" variable, which means that this input is kept constant (or, more precisely, the input is left "unused" for control in this layer).

In general, H_1 and H_2 are "full" matrices, allowing for measurement combinations as controlled variables. However, for simplicity, especially in the regulatory layer, we often pefer to control individual measurements, that is, H_2 is usually a "selection matrix," where each row in H_2 contains one 1-element (to identify the selected variable) with the remaining elements set to 0. In this case, we can write $CV_2 = H_2 y = [y_2; u_1]$, where y_2 denotes the actual controlled variables in the regulatory layer, whereas u_1 denotes the "unused" inputs (u_1), which are left as degrees of freedom for the supervisory layer. Note that this indirectly determines the inputs u_2 used in the regulatory layer to control y_2, because u_2 is what remains in the set u after selecting u_1. To have a simple control structure, with as few regulatory loops as possible, it is desirable that H_2 is selected such that there are many inputs (u_1) left "unused" in the regulatory layer.

Example. Assume there are three candidate output measurements (temperatures T) and two inputs (flowrates q),

$$y_m = [T_a T_b T_c] , \quad u = [q_a q_b]$$

and we have by definition $y = [y_m; u]$. Then the choice

$$H_2 = [0\ 1\ 0\ 0\ 0; \ 0\ 0\ 0\ 0\ 1]$$

means that we have selected $CV_2 = H_2 y = [T_b; q_b]$. Thus, $u_1 = q_b$ is an unused input for regulatory control, and in the regulatory layer we close one loop, using $u_2 = q_a$ to control $y_2 = T_b$. If we instead select

$$H_2 = [1\ 0\ 0\ 0\ 0; \ 0\ 0\ 1\ 0\ 0]$$

then we have $CV_2 = [T_a; T_c]$. None of these are inputs, so u_1 is an empty set in this case. This means that we need to close two regulatory loops, using $u_2 = [q_a; q_b]$ to control $y_2 = [T_a; T_c]$.

Supervisory Control Layer and Selection of Economic Controlled Variables (CV_1)

Some objectives for the supervisory control layer are given in Table 2. The main structural issue for the supervisory control layer, and probably the most important decision in the design of any control system, is the selection of the primary (economic) controlled variable CV_1. In many cases, a good engineer can make a reasonable choice based on process insight and experience. However, the control engineer must realize that this is a critical decision. The main rules and issues for selecting CV_1 are

CV_1 Rule 1. Control active constraints (almost always)

- Active constraints may often be identified by engineering insight, but more generally requires optimization based on a detailed model.

  For example, consider the problem of minimizing the driving time between two cities (cost $J = T$). There is a single input ($u = $ fuel flow f [l/s]) and the optimal solution is often constrained. When driving a fast car, the active constraint may be the speed limit ($CV_1 = v$ [km/h] with setpoint v_{max}, e.g., $v_{max} = 100\,km/h$). When driving

Control Structure Selection, Table 2 Objectives of supervisory control layer

O1. Control primary "economic" variables CV_1 at setpoint using as degrees of freedom MV_1, which includes the setpoints to the regulatory layer ($y_{2s} = CV_{2s}$) as well as any "unused" degrees of freedom (u_1)

O2. Switch controlled variables (CV_1) depending on operating region, for example, because of change in active constraints

O3. Supervise the regulatory layer, for example, to avoid input saturation (u_2), which may destabilize the system

O4. Coordinate control loops (multivariable control) and reduce effect of interactions (decoupling)

O5. Provide feedforward action from measured disturbances

O6. Make use of additional inputs, for example, to improve the dynamic performance (usually combined with input midranging control) or to extend the steady-state operating range (split range control)

O7. Make use of extra measurements, for example, to estimate the primary variables CV_1

an old car, the active constraint maybe the maximum fuel flow ($CV_1 = f [l/s]$ with setpoint f_{max}). The latter corresponds to an input constraint ($u_{max} = f_{max}$) which is trivial to implement ("full gas"); the former corresponds to an output constraint ($y_{max} = v_{max}$) which requires a controller ("cruise control").

- For *"hard" output constraints*, which cannot be violated at any time, we need to introduce a *backoff* (safety margin) to guarantee feasibility. The backoff is defined as the difference between the optimal value and the actual setpoint, for example, we need to back off from the speed limit because of the possibility for measurement error and imperfect control

$$CV_{1,s} = CV_{1,max} - \text{backoff}$$

For example, to avoid exceeding the speed limit of 100 km/h, we may set backoff = 5 km/h, and use a setpoint $v_s = 95$ km/h rather than 100 km/h.

CV_1 Rule 2. For the remaining unconstrained degrees of freedom, look for "self-optimizing" variables which when held constant, indirectly lead to close-to-optimal operation, in spite of disturbances.

- Self-optimizing variables ($CV_1 = H_1 y$) are variables which when kept constant, indirectly (through the action of the feedback control system) lead to close-to optimal adjustment of the inputs (u) when there are disturbances (d).

- An ideal self-optimizing variable is the gradient of the cost function with respect to the unconstrained input. $CV_1 = dJ/du = J_u$
- More generally, since we rarely can measure the gradient J_u, we select $CV_1 = H_1 y$. The selection of a good H_1 is a nontrivial task, but some quantitative approaches are given below.

For example, consider again the problem of driving between two cities, but assume that the objective is to minimize the total fuel, $J = V$ [liters]., Here, driving at maximum speed will consume too much fuel, and driving too slow is also nonoptimal. This is an unconstrained optimization problem, and identifying a good CV_1 is not obvious. One option is to maintain a constant speed ($CV_1 = v$), but the optimal value of v may vary depending on the slope of the road. A more "self-optimizing" option, could be to keep a constant fuel rate ($CV_1 = f [l/s]$), which will imply that we drive slower uphill and faster downhill. More generally, one can control combinations, $CV_1 = H_1 y$ where H_1 is a "full" matrix.

CV_1 Rule 3. For the unconstrained degrees of freedom, one should *never* control a variable that reaches its maximum or minimum value at the optimum, for example, never try to control directly the cost J. Violation of this rule gives either infeasibility (if attempting to control J at a lower value than J_{min}) or nonuniqueness (if attempting to control J at higher value than J_{min}).

Assume again that we want to minimize the total fuel needed to drive between two cities, $J = V$ [l]. Then one should avoid fixing the

total fuel, $CV_1 = V$ [l], or, alternatively, avoid fixing the fuel consumption("gas mileage") in liters pr. km ($CV_1 = f$ [l/km]). Attempting to control the fuel consumption[l/km] <u>below</u> the car's minimum value is obviously not possible (infeasible). Alternatively, attempting to control the fuel consumption <u>above</u> its minimum value has two possible solutions; driving slower or faster than the optimum. Note that the policy of controlling the fuel rate f [l/s] at a fixed value will never become infeasible.

For CV_1-Rule 2, it is always possible to find good variable combinations (i.e., H_1 is a "full" matrix), at least locally, but whether or not it is possible to find good individual variables (H_1 is a selection matrix), is not obvious. To help identify potential "self-optimizing" variables ($CV_1 = c$) ,the following requirements may be used:

Requirement 1. The *optimal* value of c is insensitive to disturbances, that is, $dc_{opt}/dd = H_1F$ is small. Here $F = dy_{opt}/dd$ is the optimal sensitivity matrix (see below).

Requirement 2. The variable c is easy to measure and control accurately

Requirement 3. The value of c is sensitive to changes in the manipulated variable, u; that is, the gain, $G = HG^y$, from u to c is large (so that even a large error in controlled variable, c, results in only a small variation in u.) Equivalently, the optimum should be "flat" with respect to the variable, c. Here $G^y = dy/du$ is the measurement gain matrix (see below).

Requirement 4. For cases with two or more controlled variables c, the selected variables should not be closely correlated.

All four requirements should be satisfied. For example, for the operation of a marathon runner, the heart rate may be a good "self-optimizing" controlled variable c (to keep at constant setpoint). Let us check this against the four requirements. The optimal heart rate is weakly dependent on the disturbances (requirement 1) and the heart rate is easy to measure (requirement 2). The heart rate is quite sensitive to changes in power input (requirement 3). Requirement 4 does not apply since this is

a problem with only one unconstrained input (the power). In summary, the heart rate is a good candidate.

Regions and switching. If the optimal active constraints vary depending on the disturbances, new controlled variables (CV_1) must be identified (offline) for each active constraint region, and online switching is required to maintain optimality. In practise, it is easy to identify when to switch when one reaches a constraint. It is less obvious when to switch out of a constraint, but actually one simply has to monitor the value of the unconstrained CVs from the neighbouring regions and switch out of the constraint region when the unconstrained CV reaches its setpoint.

In general, one would like to simplify the control structure and reduce need for switching. This may require using a suboptimal CV_1 in some regions of active constraints. In this case, the setpoint for CV_1 may not be its nominally optimal value (which is the normal choice), but rather a "robust setpoint" (with backoff) which reduces the loss when we are outside the nominal constraint region.

Structure of supervisory layer. The supervisory layer may either be centralized, e.g., using model predictive control (MPC), or decomposed into simpler subcontrollers using standard elements, like decentralized control (PID), cascade control, selectors, decouplers, feedforward elements, ratio control, split range control, and input midrange control (also known as input resetting, valve position control or habituating control). In theory, the performance is better with the centralized approach (e.g., MPC), but the difference can be small when designed by a good engineer. The main reasons for using simpler elements is that (1) the system can be implemented in the existing "basic" control system, (2) it can be implemented with little model information, and (3) it can be build up gradually. However, such systems can quickly become complicated and difficult to understand for other than the engineer who designed it. Therefore, model-based centralized solutions (MPC) are often preferred because the design is more systematic and easier to modify.

Quantitative Approach for Selecting Economic Controlled Variables, CV_1

A quantitative approach for selecting economic controlled variables is to consider the effect of the choice $CV_1 = H_1y$ on the economic cost J when disturbances d occur. One should also include noise/errors (n^y) related to the measurements and inputs.

Step S1. *Define operational objectives (economic cost function J and constraints)*

We first quantify the operational objectives in terms of a scalar cost function J [\$/s] that should be minimized (or equivalently, a scalar profit function, $P = -J$, that should be maximized). For process control applications, this is usually easy, and typically we have

$$J = \text{cost feed} + \text{cost utilities (energy)}$$
$$- \text{value products [\$/s]}$$

Note that the economic cost function J is used to *select* the controlled variables (CV_1), and another cost function (J'), typically involving the deviation in CV_1 from their optimal setpoints CV_{1s}, is used for the actual controller design (e.g., using MPC).

Step S2. *Find optimal operation for expected disturbances*

Mathematically, the optimization problem can be formulated as

$$\min_u J(u, x, d)$$

subject to:

Model equations: $dx/dt = f(u, x, d)$
Operational constraints: $g(u, x, d) \leq 0$

In many cases, the economics are determined by the steady-state behavior, so we can set $dx/dt = 0$. The optimization problem should be resolved for the expected disturbances (d) to find the truly optimal operation policy, $u_{opt}(d)$. The nominal solution (d_{nom}) may be used to obtain the setpoints (CV_{1s}) for the selected controlled variables. In

practise, the optimum input $u_{opt}(d)$ cannot be realized, because of model error and unknown disturbances d, so we use a feeback implementation where u is adjusted to keep the selected variables CV_1 at their nominally optimal setpoints.

Together with obtaining the model, the optimization step S2 is often the most time consuming step in the entire plantwide control procedure.

Step S3. *Select supervisory (economic) controlled variables, CV_1*

CV_1-Rule 1: Control Active Constraints

A primary goal for solving the optimization problem is to find the expected regions of active constraints, and a constraint is said to be "active" if $g = 0$ at the optimum. The optimally active constraints will vary depending on disturbances (d) and market conditions (prices).

CV_1-Rule 2: Control Self-Optimizing Variables

After having identified (and controlled) the active constraints, one should consider the remaining lower-dimension unconstrained optimization problem, and for the remaining unconstrained degrees of freedom one should search for *control "self-optimizing" variables* c.

1. **"Brute force" approach.** Given a set of controlled variables $CV_1 = c = H_1y$, one computes the cost J(c,d) when we keep c constant ($c = c_s + H_1n^y$) for various disturbances (d) and measurement errors (n^y). In practise, this is done by running a large number of steady-state simulations to try to cover the expected future operation.

2. **"Local" approaches** based on a quadratic approximation of the cost J. Linear models are used for the effect of u and d on y.

$$y = G^yu + G_d^yd$$

This is discussed in more detail in Alstad et al. (2009) and references therein. The main local approaches are:

2A. **Maximum gain rule: maximize the minimum singular value of $G = H_1G^y$.**

In other words, the maximum gain rule, which essentially is a quantitative version of Requirements 1, 3 and 4 given above, says that one should control "sensitive" variables, with a large scaled gain G from the inputs (u) to $c = H_1 y$. This rule is good for pre-screening and also yields good insight.

2B. **Nullspace method.** This method yields optimal measurement combinations for the case with no noise, $n^y = 0$. One must first obtain the optimal measurement sensitivity matrix F, defined as

$$\mathbf{F} = dy^{opt}/dd.$$

Each column in **F** expresses the optimal change in the y's when the independent variable (u) is adjusted so that the system remains optimal with respect to the disturbance d. Usually, it is simplest to obtain F numerically by optimizing the model. Alternatively, we can obtain F from a quadratic approximation of the cost function

$$F = G_d^y - G^y J_{uu}^{-1} J_{ud}$$

Then, assuming that we have at least as many (independent) measurements y as the sum of the number of (independent) inputs (u) and disturbances (d), the optimal is to select $c = H_1 y$ such that

$$H_1 \mathbf{F} = 0$$

Note that H_1 is a nonsquare matrix, so $H_1 F = 0$ does not require that $H_1 = 0$ (which is a trivial uninteresting solution), but rather that H_1 is in the nullspace of F^T.

2C. **Exact local** method (loss method). This extends the nullspace method to include noise (n^y) and allows for any number of measurements. The noise and disturbances are normalized by introducing weighting matrices W_{ny} and W_d (which have the expected magnitudes along the diagonal) and then the expected loss, $L = J - J_{opt}(d)$, is minimized by selecting H_1 to solve the following problem

$$\min\_H_1 ||M(H_1)||_2$$

where 2 denotes the Frobenius norm and

$$M(H_1) = J_{uu}^{1/2}(H_1 G^y)^{-1} H_1 Y, Y$$
$$= [FW_d \ W_{ny}].$$

Note here that the optimal choice with $W_{ny} = 0$ (no noise) is to choose H_1 such that $H_1 F = 0$, which is the nullspace method. For the general case, when H_1 is a "full" matrix, this is a convex problem and the optimal solution is $H_1^T = (YY')^{-1} G^y Q$ where Q is any nonsingular matrix.

Regulatory Control Layer

The main purpose of the regulatory layer is to "stabilize" the plant, preferably using a *simple* control structure (e.g., single-loop PID controllers) which does not require changes during operation. "Stabilize" is here used in a more extended sense to mean that the process does not "drift" too far away from acceptable operation when there are disturbances. The regulatory layer should make it possible to use a "slow" supervisory control layer that does not require a detailed model of the high-frequency dynamics. Therefore, in addition to track the setpoints given by the supervisory layer (e.g., MPC), the regulatory layer may directly control primary variables (CV_1) that require fast and tight control, like economically important active constraints.

In general, the design of the regulatory layer involves the following structural decisions:

1. Selection of controlled outputs y_2 (among all candidate measurements y_m).
2. Selection of inputs $MV_2 = u_2$ (a subset of all available inputs u) to control the outputs y_2.
3. Pairing of inputs u_2 and outputs y_2 (since decentralized control is normally used).

Decisions 1 and 2 combined (IO selection) is equivalent to selecting H_2 (Fig. 3). Note that we do not "use up" any degrees of freedom in the regulatory layer because the set points (y_{2s})

become manipulated variables (MV_1) for the supervisory layer (see Fig. 3). Furthermore, since the set points are set by the supervisory layer in a cascade manner, the system eventually approaches the same steady-state (as defined by the choice of economic variables CV_1) regardless of the choice of controlled variables in the regulatory layer.

The inputs for the regulatory layer (u_2) are selected as a subset of all the available inputs (u). For stability reasons, one should avoid input saturation in the regulatory layer. In particular, one should avoid using inputs (in the set u_2) that are optimally constrained in some disturbance region. Otherwise, in order to avoid input saturation, one needs to include a backoff for the input when entering this operational region, and doing so will have an economic penalty.

In the regulatory layer, the outputs (y_2) are usually selected as individual measurements and they are often not important variables in themselves. Rather, they are "extra outputs" that are controlled in order to "stabilize" the system, and their setpoints (y_{2s}) are changed by the layer above, to obtain economical optimal operation. For example, in a distillation column one may control a temperature somewhere in the middle of the column ($y_2 = T$) in order to "stabilize" the column profile. Its setpoint ($y_{2s} = T_s$) is adjusted by the supervisory layer to obtain the desired product composition ($y_1 = c$).

Input-Output (IO) Selection for Regulatory Control (u_2, y_2)

Finding the truly optimal control structure, including selecting inputs and outputs for regulatory control, requires finding also the optimal controller parameters. This is an extremely difficult mathematical problem, at least if the controller K is decomposed into smaller controllers. In this section, we consider some approaches which does not require that the controller parameters be found. This is done by making assumptions related to achievable control performance (controllability) or perfect control.

Before we look at the approaches, note again that the IO-selection for regulatory control may be combined into a single decision, by considering the selection of

$$CV_2 = [y_2; u_1] = H_2 y$$

Here u_1 denotes the inputs that are <u>not</u> used by the regulatory control layer. This follows because we want to use all inputs u for control, so assuming that the set u is given, "selection of inputs u_2" (decision 2) is by elimination equivalent to "selection of inputs u_1." Note that CV_2 include all variables that we keep at desired (constant) values within the fast time horizon of the regulatory control layer, including the "unused" inputs u_1

Survey by Van de Wal and Jager

Van de Wal and Jager provide an overview of methods for input-output selection, some of which include:

1. "Accessibility" based on guaranteeing a cause–effect relationship between the selected inputs (u_2) and outputs (y_2). Use of such measures may eliminate unworkable control structures.
2. "State controllability and state observability" to ensure that any unstable modes can be stabilized using the selected inputs and outputs.
3. "Input-output controllability" analysis to ensure that y_2 can be acceptably controlled using u_2. This is based on scaling the system, and then analysing the transfer matrices $G_2(s)$ (from u_2 to y_2) and G_{d2} (from expected disturbances d to y_2). Some important controllability measures are right half plane zeros (unstable dynamics of the inverse), condition number, singular values, relative gain array, etc. One problem here is that there are many different measures, and it is not clear which should be given most emphasis.
4. "Achievable robust performance." This may be viewed as a more detailed version of input-output controllability, where several relevant issues are combined into a single measure. However, this requires that the control problem can actually be formulated clearly, which may be very difficult, as already mentioned.

In addition, it requires finding the optimal robust controller for the given problem, which may be very difficult.

Most of these methods are useful for analyzing a given structure (u_2, y_2) but less suitable for selection. Also, the list of methods is also incomplete, as disturbance rejection, which is probably the most important issue for the regulatory layer, is hardly considered.

A Systematic Approach for IO-Selection Based on Minimizing State Drift Caused by Disturbances

The objectives of the regulatory control layer are many, and Yelchuru and Skogestad (2013) list 13 partly conflicting objectives. To have a truly systematic approach to regulatory control design, including IO-selection, we would need to quantify all these partially conflicting objectives in terms of a scalar cost function J_2. We here consider a fairly general cost function,

$$J_2 = ||Wx||$$

which may be interpreted as the weighted state drift. One justification for considering the state drift, is that the regulatory layer should ensure that the system, as measured by the weighted states Wx, does not drift too far away from the desired state, and thus stays in the "linear region" when there are disturbances. Note that the cost J_2 is used to *select* controlled variables (CV_2) and not to design the controller (for which the cost may be the control error, $J_2' = ||CV_2 - CV_{2s}||$).

Within this framework, the IO-selection problem for the regulatory layer is then to select the nonsquare matrix H_2,

$$CV_2 = H_2y$$

where $y = [y_m; u]$, such that the cost J_2 is minimized. The cause for changes in J_2 are disturbances d, and we consider the linear model (in deviation variables)

$$y = G^y u + G_d^y d$$
$$x = G^x u + G_d^x d$$

where the G-matrices are transfer matrices. Here, G_d^x gives the effect of the disturbances on the states with no control, and the idea is to reduce the disturbance effect by closing the regulatory control loops. Within the "slow" time scale of the supervisory layer, we can assume that CV_2 is perfectly controlled and thus constant, or $CV_2 = 0$ in terms of deviation variables. This gives

$$CV_2 = H_2G^y u + H_2G_d^y d = 0$$

and solving with respect to u gives

$$u = - (H_2G^y)^{-1} (H_2G_d^y) d$$

and we have

$$x = P_d^x (H_2) d$$

where

$$P_d^x (H_2) = G_d^x - G^x (H_2G^y)^{-1} H_2G_d^y$$

is the disturbance effect for the "partially" controlled system with only the regulatory loops closed. Note that it is not generally possible to make $P_d^x = 0$ because we have more states than we have available inputs. To have a small "state drift," we want $J_2 = ||W P_d d||$ to be small, and to have a simple regulatory control system we want to close as few regulatory loops as possible. Assume that we have normalized the disturbances so that the norm of d is 1, then we can solve the following problem

For $0, 1, 2\ldots$ etc. loops closed solve: min_H_2||$M_2 (H_2)$ ||

where $M_2 = WP_d^x$ and $\dim (u2) = \dim (y2) =$ no. of loops closed.

By comparing the value of $||M_2 (H_2)||$ with different number of loops closed (i.e., with different H_2), we can then decide on an appropriate regulatory layer structure. For example, assume that we find that the value of J_2 is 110 (0 loops closed), 0.2 (1 loop), and 0.02 (2 loops), and assume we have scaled the disturbances and states such that a J_2-value less than about 1 is acceptable, then closing 1 regulatory loop is probably the best choice.

In principle, this is straightforward, but there are three remaining *issues*: (1) We need to choose an appropriate norm, (2) we should include measurement noise to avoid selecting insensitive measurements and (3) the problem must be solvable numerically.

Issue 1. The norm of M_2 should be evaluated in the frequency range between the "slow" bandwidth of the supervisory control layer (ω_{B1}) and the "fast" bandwidth of the regulatory control layer (ω_{B2}). However, since it is likely that the system sometimes operates without the supervisory layer, it is reasonable to evaluate the norm of P_d^x in the frequency range from 0 (steady state) to ω_{B2}. Since we want H_2 to be a constant (not frequency-dependent) matrix, it is reasonable to choose H_2 to minimize the norm of M_2 at the frequency where $||M_2||$ is expected to have its peak. For some mechanical systems, this may be at some resonance frequency, but for process control applications it is usually at steady state ($\omega = 0$), that is, we can use the steady-state gain matrices when computing P_d^x. In terms of the norm, we use the 2-norm (Frobenius norm), mainly because it has good numerical properties, and also because it has the interpretation of giving the expected variance in x for normally distributed disturbances.

Issues 2 and 3. If we include also measurement noise n^y, which we should, then the expected value of J_2 is minimized by solving the problem min_H_2 $||M_2(H_2)||_2$ where (Yelchuru and Skogestad 2013)

$$\mathbf{M}_2(\mathbf{H}_2) = \mathbf{J}_{2uu}^{1/2}(\mathbf{H}_2\mathbf{G}^y)^{-1}\mathbf{H}_2\mathbf{Y}_2$$

$$\mathbf{Y}_2 = [\mathbf{F}_2\mathbf{W}_d \quad \mathbf{W}_n]; \quad \mathbf{F}_2 = \frac{\partial \mathbf{y}_{opt}}{\partial \mathbf{d}}$$
$$= \mathbf{G}^y \mathbf{J}_{2uu}^{-1}\mathbf{J}_{2ud} - \mathbf{G}_{dy}$$

where $J_{2uu} \stackrel{\Delta}{=} \frac{\partial^2 J_2}{\partial u^2} = 2\mathbf{G}^{xT}\mathbf{W}^T\mathbf{W}\mathbf{G}^x$, $J_{2ud} \stackrel{\Delta}{=} \frac{\partial^2 J_2}{\partial u \partial d} = 2\mathbf{G}^{xT}\mathbf{W}^T\mathbf{W}\mathbf{G}_{dx}$,

Note that this is the same mathematical problem as the "exact local method" presented for selecting $CV_1 = H_1 y$ for minimizing the economic cost J, but because of the specific simple form

for the cost J_2, it is possible to obtain analytical formulas for the optimal sensitivity, F_2. Again, W_d and W_{ny} are diagonal matrices, expressing the expected magnitude of the disturbances (d) and noise (for y).

For the case when H_2 is a "full" matrix, this can be reformulated as a convex optimization problem and an explicit solution is

$$\mathbf{H}_2^T = (\mathbf{Y}_2\mathbf{Y}_2^T)^{-1}\mathbf{G}^y(\mathbf{G}^{yT}(\mathbf{Y}_2\mathbf{Y}_2^T)^{-1}\mathbf{G}^y)^{-1}\mathbf{J}_{2uu}^{1/2}$$

and from this we can find the optimal value of J_2. It may seem restrictive to assume that H_2 is a "full" matrix, because we usually want to control individual measurements, and then H_2 should be a selection matrix, with 1's and 0's. Fortunately, since we in this case want to control as many measurements (y_2) as inputs (u_2), we have that H_2 is square in the selected set, and the optimal value of J_2 when H_2 is a selection matrix is the same as when H_2 is a full matrix. The reason for this is that specifying (controlling) any linear combination of y_2, uniquely determines the individual y_2's, since dim(u_2) = dim(y_2). Thus, we can find the optimal selection matrix H_2, by searching through all the candidate square sets of y. This can be effectively solved using the branch and bound approach of Kariwala and Cao, or alternatively it can be solved as a mixed-integer problem with a quadratic program (QP) at each node (Yelchuru and Skogestad 2012). The approach of Yelchuru and Skogestad can also be applied to the case where we allow for disjunct sets of measurement combinations, which may give a lower J_2 in some cases.

Comments on the state drift approach.
1. We have assumed that we perfectly control y_2 using u_2, at least within the bandwidth of the regulatory control system. Once one has found a candidate control structure (H_2), one should check that it is possible to achieve acceptable control. This may be done by analyzing the input-output controllability of the system $y_2 = G_2u_2 + G_{2d}$ d, based on the transfer matrices $G_2 = H_2G^y$ and $G_{2d} = H_2G_d^y$. If the controllability of this system is not acceptable, then

one should consider the second-best matrix H_2 (with the second-best value of the state drift J_2) and so on.

2. The state drift cost drift $J_2 = ||Wx||$ is in principle independent of the economic cost (J). This is an advantage because we know that the economically optimal operation (e.g., active constraints) may change, whereas we would like the regulatory layer to remain unchanged. However, it is also a disadvantage, because the regulatory layer determines the initial response to disturbances, and we would like this initial response to be in the right direction economically, so that the required correction from the slower supervisory layer is as small as possible. Actually, this issue can be included by extending the state vector x to include also the economic controlled variables, CV_1, which is selected based on the economic cost J. The weight matrix W may then be used to adjust the relative weights of avoiding drift in the internal states x and economic controlled variables CV_1.

3. The above steady-state approach does not consider input-output pairing, for which dynamics are usually the main issue. The main pairing rule is to "pair close" in order to minimize the effective time delay between the selected input and output. For a more detailed approach, decentralized input-output controllability must be considered.

Summary and Future Directions

Control structure design involves the structural decisions that must be made before designing the actual controller, and it is in most cases a much more important step than the controller design. In spite of this, the theoretical tools for making the structural decisions are much less developed than for controller design. This chapter summarizes some approaches, and it is expected, or at least hoped, that this important area will further develop in the years to come.

The most important structural decision is usually related to selecting the economic controlled variables, $CV_1 = H_1y$, and the stabilizing controlled variables, $CV_2 = H_2y$. However, control engineers have traditionally not used the degrees of freedom in the matrices H_1 and H_2, and this chapter has summarized some approaches.

There has been a belief that the use of "advanced control," e.g., MPC, makes control structure design less important. However, this is not correct because also for MPC must one choose inputs ($MV_1 = CV_{2s}$) and outputs (CV_1). The selection of CV_1 may to some extent be avoided by use of "Dynamic Real-Time Optimization (DRTO)" or "Economic MPC," but these optimizing controllers usually operate on a slower time scale by sending setpoints to the basic control layer ($MV_1 = CV_{2s}$), which means that selecting the variables CV_2 is critical for achieving (close to) optimality on the fast time scale.

Cross-References

▶ Control Hierarchy of Large Processing Plants: An Overview
▶ Industrial MPC of Continuous Processes
▶ PID Control

Bibliography

Alstad V, Skogestad S (2007) Null space method for selecting optimal measurement combinations as controlled variables. Ind Eng Chem Res 46(3): 846–853

Alstad V, Skogestad S, Hori ES (2009) Optimal measurement combinations as controlled variables. J Process Control 19:138–148

Downs JJ, Skogestad S (2011) An industrial and academic perspective on plantwide control. Ann Rev Control 17:99–110

Engell S (2007) Feedback control for optimal process operation. J Proc Control 17:203–219

Foss AS (1973) Critique of chemical process control theory. AIChE J 19(2):209–214

Kariwala V, Cao Y (2010) Bidirectional branch and bound for controlled variable selection. Part III. Local average loss minimization. IEEE Trans Ind Inform 6: 54–61

Kookos IK, Perkins JD (2002) An Algorithmic method for the selection of multivariable process control structures. J Proc Control 12:85–99

Morari M, Arkun Y, Stephanopoulos G (1973) Studies in the synthesis of control structures for chemical processes. Part I. AIChE J 26:209–214

Narraway LT, Perkins JD (1993) Selection of control structure based on economics. Comput Chem Eng 18:S511–S515

Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. J Proc Control 10:487–507

Skogestad S (2004) Control structure design for complete chemical plants. Comput Chem Eng 28(1–2):219–234

Skogestad S (2012) Economic plantwide control, chapter 11. In: Rangaiah GP, Kariwala V (eds) Plantwide control. Recent developments and applications. Wiley, Chichester, pp 229–251. ISBN:978-0-470-98014-9

Skogestad S, Postlethwaite I (2005) Multivariable feedback control, 2nd edn. Wiley, Chichester

van de Wal M, de Jager B (2001) Review of methods for input/output selection. Automatica 37:487–510

Yelchuru R, Skogestad S (2012) Convex formulations for optimal selection of controlled variables and measurements using Mixed Integer Quadratic Programming. J Process Control 22:995–1007

Yelchuru R, Skogestad S (2013) Quantitative methods for regulatory layer selection. J Process Control 23:58–69

Controllability and Observability

H.L. Trentelman
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Abstract

State controllability and observability are key properties in linear input–output systems in state-space form. In the state-space approach, the relation between inputs and outputs is represented using the state variables of the system. A natural question is then to what extent it is possible to manipulate the values of the state vector by means of an appropriate choice of the input function. The concepts of controllability, reachability, and null controllability address this issue. Another important question is whether it is possible to uniquely determine the values of the state vector from knowledge of the input and output

signals over a given time interval. This question is dealt with using the concept of observability.

Keywords

Introduction

In the state-space approach to input–output systems, the relation between input signals and output signals is represented by means of two equations. In the continuous-time case, the first of these equations is a first-order vector differential equation driven by the input signal and is often called *the state equation*. The second equation is an algebraic equation, often called the *output equation*. The unknown in the differential equation is called the *state vector* of the system. Given a particular input signal and initial value of the state vector, the state equation generates a unique solution, called the state trajectory of the system. The output equation determines the corresponding output signal as a function of this state trajectory and the input signal. Thus, in the state space approach, the input–output behavior of the system is obtained using the state vector as an intermediate variable.

In the context of input–output systems in state-space form, the properties of controllability and observability characterize the interaction between the input, the state, and the output. In particular, controllability describes the ability to manipulate the state vector of the system by applying appropriate input signals. Observability describes the ability to determine the values of the state vector from knowledge of the input and output over a certain time interval. The properties of controllability and observability are fundamental properties that play a major role in the analysis and control of linear input–output systems in state-space form.

Systems with Inputs and Outputs

Consider a continuous-time, linear, time-invariant, input–output system in state-space form represented by the equations

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t) + Du(t). \qquad (1)$$

This system is referred to as Σ. In Eq. (1), A, B, C, and D are maps (or matrices), and the functions x, u, and y are considered to be defined on the real axis $\mathbb{R}$ or on any subinterval of it. In particular, one often assumes the domain of definition to be the nonnegative part of $\mathbb{R}$, which is without loss of generality since the system is time-invariant. The function u is called the *input*, and its values are assumed to be given. The class of admissible input functions is denoted by **U**. Often, **U** is the class of piecewise continuous or locally integrable functions, but for most purposes, the exact class from which the input functions are chosen is not important. We assume that input functions take values in an m-dimensional space $\mathcal{U}$, which we often identify with $\mathbb{R}^m$. The first equation of Σ is an ordinary differential equation for the variable x. For a given initial value of x and input function u, the function x is completely determined by this equation. The variable x is called the *state variable* and it is assumed to take values in an n-dimensional space $\mathcal{X}$. The space $\mathcal{X}$ is called the *state space*. It is usually identified with $\mathbb{R}^n$. Finally, y is called the *output* of the system and takes values in a p-dimensional space $\mathcal{Y}$, which we identify with $\mathbb{R}^p$. Since the system Σ is completely determined by the maps (or matrices) A, B, C, and D, we identify Σ with the quadruple (A, B, C, D).

The solution of the differential equation of Σ with initial value $x(0) = x_0$ is denoted as $x_u(t, x_0)$. It can be given explicitly using the variation-of-constants formula, namely,

$$x_u(t, x_0) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau. \quad (2)$$

The corresponding value of y is denoted by $y_u(t, x_0)$. As a consequence of (2), we have

$$y_u(t, x_0) = Ce^{At}x_0 + \int_0^t K(t - \tau)u(\tau)\,\mathrm{d}\tau$$
$$+ Du(t), \qquad (3)$$

where $K(t) := Ce^{At}B$. In the case $D = 0$, it is customary to call $K(t)$ the *impulse response*. In the general case, one would call the distribution $K(t) + D\delta(t)$ the impulse response.

Controllability

Controllability is concerned with the ability to manipulate the state by choosing an appropriate input signal, thus steering the current state to a desired future state in a given finite time. Thus, in particular, in the differential equation in (1), we study the relation between u and x. We investigate to what extent one can influence the state x by a suitable choice of the input u.

For this purpose, we introduce the (at time T) *reachable space* $\mathcal{W}_T$, defined as the space of points x_1 for which there exists an input u such that $x_u(T, 0) = x_1$, i.e., the set of points that can be reached from the origin at time T. It follows from the linearity of the differential equation that $\mathcal{W}_T$ is a linear subspace of $\mathcal{X}$. In fact, (2) implies

$$\mathcal{W}_T = \left\{ \int_0^T e^{A(T-\tau)}Bu(\tau)\mathrm{d}\tau \,\middle|\, u \in \mathbf{U} \right\}. \quad (4)$$

We call system Σ *reachable at time T* if every point can be reached from the origin, i.e., if $\mathcal{W}_T = \mathcal{X}$. It follows from (2) that if the system is reachable at time T, every point can be reached from every point at time T, because the condition for the point x_1 to be reachable from x_0 at time T is

$$x_1 - e^{AT}x_0 \in \mathcal{W}_T.$$

The property that every point is reachable from any point in a given time interval [0, T] is called *controllability (at T)*. Finally, we have the concept of *null controllability*, i.e., the possibility to reach the origin from an arbitrary initial point. According to (2), for a point x_0 to be null controllable at T, we must have

$$e^{AT} x_0 + \int_0^T e^{A(T-\tau)} B u(\tau)\, \mathrm{d}\tau = 0$$

for some $u \in \mathbf{U}$. We observe that x_0 is null controllable at T (by the control u) if and only if $-e^{AT} x_0$ is reachable at T (by the control u). Since e^{AT} is invertible, we see that Σ is null controllable at T if and only if Σ is reachable at T. Henceforth, we refer to the equivalent properties reachability, controllability, null controllability simply as controllability (at T). It should be remarked that the equivalence of these concepts does not hold in other situations, e.g., for discrete-time systems. We intend to obtain an explicit expression for the space $\mathcal{W}_T$ and, based on this, an explicit condition for controllability. This is provided by the following result.

Theorem 1 *Let η be an n-dimensional row vector and $T > 0$. Then the following statements are equivalent:*
1. *$\eta \perp \mathcal{W}_T$ (i.e., $\eta x = 0$ for all $x \in \mathcal{W}_T$).*
2. *$\eta e^{tA} B = 0$ for $0 \le t \le T$.*
3. *$\eta A^k B = 0$ for $k = 0, 1, 2, \ldots$.*
4. *$\eta (B\ AB\ \cdots\ A^{n-1} B) = 0$.*

Proof (i) $\Leftrightarrow$ (ii) If $\eta \perp \mathcal{W}_T$, then by Eq. (4):

$$\int_0^T \eta e^{A(T-\tau)} B u(\tau)\, \mathrm{d}\tau = 0 \qquad (5)$$

for every $u \in \mathbf{U}$. Choosing $u(t) = B^{\mathrm{T}} e^{A^{\mathrm{T}}(T-t)} \eta^{\mathrm{T}}$ for $0 \le t \le T$ yields

$$\int_0^T \left\| \eta e^{A(T-\tau)} B \right\|^2 \mathrm{d}\tau = 0,$$

from which (ii) follows. Conversely, assume that (ii) holds. Then (5) holds and hence (i) follows.

(ii) $\Leftrightarrow$ (iii) This is obtained by power series expansion of e^{At} $\left(= \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k\right)$.

(iii) $\Leftrightarrow$ (iv) This follows immediately from the evaluation of the vector-matrix product.

(iv) $\Leftrightarrow$ (iii) This implication is based on the Cayley-Hamilton Theorem. According to this theorem, A^n is a linear combination of $I, A, \ldots, A^{n-1}$. By induction, it follows

that A^k $(k > n)$ is a linear combination of $I, A, \ldots, A^{n-1}$ as well. Therefore, $\eta A^k B = 0$ for $k = 0, 1, \ldots, n - 1$ implies that $\eta A^k B = 0$ for all $k \in \mathbb{N}$. $\square$

As an immediate consequence of the previous theorem, we find that at time T reachable subspace $\mathcal{W}_T$ can be expressed in terms of the maps A and B as follows.

Corollary 1

$$\mathcal{W}_T = \mathrm{im}\,(B\ \ AB\ \ \cdots\ \ A^{n-1} B).$$

This implies that, in fact, $\mathcal{W}_T$ is independent of T, for $T > 0$. Because of this, we often use $\mathcal{W}$ instead of $\mathcal{W}_T$ and call this subspace the reachable subspace of Σ. This subspace of the state space has the following geometric characterization in terms of the maps A and B.

Corollary 2 *$\mathcal{W}$ is the smallest A-invariant subspace containing $\mathcal{B} := \mathrm{im}\,B$. Explicitly, $\mathcal{W}$ is A-invariant, $\mathcal{B} \subset \mathcal{W}$, and any A-invariant subspace $\mathcal{L}$ satisfying $\mathcal{B} \subset \mathcal{L}$ also satisfies $\mathcal{W} \subset \mathcal{L}$. We denote the smallest A-invariant subspace containing $\mathcal{B}$ by $\langle A | \mathcal{B} \rangle$, so that we can write $\mathcal{W} = \langle A | \mathcal{B} \rangle$. For the space $\langle A | \mathcal{B} \rangle$, we have the following explicit formula*
$$\langle A | \mathcal{B} \rangle = \mathcal{B} + A\mathcal{B} + \cdots + A^{n-1}\mathcal{B}.$$

Corollary 3 *The following statements are equivalent.*
1. *There exists $T > 0$ such that system Σ is controllable at T.*
2. *$\langle A | \mathcal{B} \rangle = \mathcal{X}$.*
3. *Rank $(B\ AB \cdots A^{n-1} B) = n$.*
4. *The system Σ is controllable at T for all $T > 0$.*

We say that the matrix pair (A, B) is *controllable* if one of these equivalent conditions is satisfied.

Example 1 Let A and B be defined by

$$A := \begin{pmatrix} -2 & -6 \\ 2 & 5 \end{pmatrix}, \qquad B := \begin{pmatrix} -3 \\ 2 \end{pmatrix}.$$

Then $(B\ AB) = \begin{pmatrix} -3 & -6 \\ 2 & 4 \end{pmatrix}$, $\mathrm{rank}(B\ AB) = 1$, and consequently, (A, B) is not controllable. The reachable subspace is the span of $(B\ AB)$, i.e., the

line given by the equation $2x_1 + 3x_2 = 0$. This can also be seen as follows. Let $z := 2x_1 + 3x_2$, then $\dot{z} = z$. Hence, if $z(0) = 0$, which is the case if $x(0) = 0$, we must have $z(t) = 0$ for all $t \geq 0$.

Observability

In this section, we include the second of equations (1), $y = Cx + Du$, in our considerations. Specifically, we investigate to what extent it is possible to reconstruct the state x if the input u and the output y are known. The motivation is that we often can measure the output and prescribe (and hence know) the input, whereas the state variable is *hidden*.

Definition 2 Two states x_0 and x_1 in $\mathcal{X}$ are called *indistinguishable* on the interval $[0, T]$ if for any input u we have $y_u(t, x_0) = y_u(t, x_1)$, for all $0 \leq t \leq T$.

Hence, x_0 and x_1 are indistinguishable if they give rise to the same output values for every input u. According to (3), for x_0 and x_1 to be indistinguishable on $[0, T]$, we must have that

$$
Ce^{At}x_0 + \int_0^t K(t - \tau) u(\tau) \, d\tau + Du(t)
$$
$$
= Ce^{At}x_1 + \int_0^t K(t - \tau) u(\tau) \, d\tau + Du(t)
$$

for $0 \leq t \leq T$ and for any input signal u. We note that the input signal does not affect distinguishability, i.e., if one u is able to distinguish between two states, then any input is. In fact, x_0 and x_1 are indistinguishable if and only if $Ce^{At}x_0 = Ce^{At}x_1$ ($0 \leq t \leq T$). Obviously, x_0 and x_1 are indistinguishable if and only if $v := x_0 - x_1$ and 0 are indistinguishable. By applying Theorem 1 with $\eta = v^T$ nonzero and transposing the equations, it follows that $Ce^{At}x_0 = Ce^{At}x_1$ ($0 \leq t \leq T$) if and only if $Ce^{At}v = 0$ ($0 \leq t \leq T$) and hence if and only if $CA^k v = 0$ ($k = 0, 1, 2, \ldots$). The Cayley-Hamilton Theorem implies that we need to consider the first n terms only, i.e.,

$$
\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} v = 0. \tag{6}
$$

As a consequence, the distinguishability of two vectors does not depend on T. The space of vectors v for which (6) holds is denoted $\langle \ker C \,|\, A \rangle$ and called the *unobservable subspace*. It is equivalently characterized as the intersection of the spaces $\ker CA^k$ for $k = 0, \ldots, n-1$, i.e.,

$$
\langle \ker C \,|\, A \rangle = \bigcap_{k=0}^{n-1} \ker CA^k.
$$

Equivalently, $\langle \ker C \,|\, A \rangle$ is the largest A-invariant subspace contained in $\ker C$. Finally, another characterization is "$v \in \langle \ker C \,|\, A \rangle$ if and only if $y_0(t, v)$ is identically zero," where the subscript "0" refers to the zero input.

Definition 3 System Σ is called *observable* if any two distinct states are not indistinguishable.

The previous considerations immediately lead to the result.

Theorem 2 *The following statements are equivalent.*
1. *The system Σ is observable.*
2. *Every nonzero state is not indistinguishable from the origin.*
3. $\langle ker\, C \,|\, A \rangle = 0$.
4. $Ce^{At}v = 0$ ($0 \leq t \leq T$) $\Rightarrow v = 0$.
5. *Rank* $\begin{pmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{pmatrix} = n$.

Since observability is completely determined by the matrix pair (C, A), we will say "(C, A) is observable" instead of "system Σ is observable."

There is a remarkable relation between the controllability and observability properties, which is referred to as *duality*. This property is most conspicuous from the conditions (3) in Corollary 3 and (5) in Theorem 2, respectively.

Specifically, (C, A) is observable if and only if (A^T, C^T) is controllable. As a consequence of duality, many theorems on controllability can be translated into theorems on observability and vice versa by mere transposition of matrices.

Example 2 Let

$$A := \begin{pmatrix} -11 & 3 \\ -3 & -5 \end{pmatrix}, \quad B := \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$
$$C := (1 \quad -1),$$

Then

$$\mathrm{rank} \begin{pmatrix} C \\ CA \end{pmatrix} = \mathrm{rank} \begin{pmatrix} 1 & -1 \\ -8 & 8 \end{pmatrix} = 1,$$

hence, (C, A) is not observable. Notice that if $v \in \langle \ker C \mid A \rangle$ and $u = 0$, identically, then $y = 0$, identically. In this example, $\langle \ker C \mid A \rangle$ is the span of $(1, 1)^T$.

Summary and Future Directions

The property of controllability can be tested by means of a rank test on a matrix involving the maps A and B appearing in the state equation of the system. Alternatively, controllability is equivalent to the property that the reachable subspace of the system is equal to the state space. The property of observability allows a rank test on a matrix involving the maps A and C appearing in the system equations. An alternative characterization of this property is that the unobservable subspace of the system is equal to the zero subspace. Concepts of controllability and observability have also been defined for discrete-time systems and, more generally, for time-varying systems and for continuous-time and discrete-time nonlinear systems.

Cross-References

▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions

▶ Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions
▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions
▶ Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions
▶ Realizations in Linear Systems Theory

Recommended Reading

The description of linear systems in terms of a state space representation was particularly stressed by R. E. Kalman in the early 1960s (see Kalman 1960a,b, 1963), Kalman et al. (1963). See also Zadeh and Desoer (1963) and Gilbert (1963). In particular, Kalman introduced the concepts of controllability and observability and gave the conditions expressed in Corollary 3, time (3), and Theorem 5, item (5). Alternative conditions for controllability and observability have been introduced in Hautus (1969) and independently by a number of authors; see Popov (1966) and Popov (1973). Other references are Belevitch (1968) and Rosenbrock (1970).

Bibliography

Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston

Belevitch V (1968) Classical network theory. Holden-Day, San Francisco

Gilbert EG (1963) Controllability and observability in multivariable control systems. J Soc Ind Appl Math A 2:128–151

Hautus MLJ (1969) Controllability and observability conditions of linear autonomous systems. Proc Nederl Akad Wetensch A 72(5):443–448

Kalman RE (1960a) Contributions to the theory of optimal control. Bol Soc Mat Mex 2:102–119

Kalman RE (1960b) On the general theory of control systems. In: Proceedings of the first IFAC congress, London: Butterworth, pp 481–491

Kalman RE (1963) Mathematical description of linear dynamical systems. J Soc Ind Appl Math A 1:152–192

Kalman RE, Ho YC, Narendra KS (1963) Controllability of linear dynamical systems. Contrib Diff Equ 1:189–213

Popov VM (1966) Hiperstabilitatea sistemelor automate. Editura Axcademiei Republicii Socialiste România (in Rumanian)

Popov VM (1973) Hyperstability of control systems. Springer, Berlin. (Translation of the previous reference)

Rosenbrock HH (1970) State-space and multivariable theory. Wiley, New York

Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London

Wonham WM (1979) Linear multivariable control: a geometric approach. Springer, New York

Zadeh LA, Desoer CA (1963) Linear systems theory – the state-space approach. McGraw-Hill, New York

Controller Performance Monitoring

Sirish L. Shah
Department of Chemical and Materials
Engineering, University of Alberta Edmonton,
Edmonton, AB, Canada

Abstract

Process control performance is a cornerstone of operational excellence in a broad spectrum of industries such as refining, petrochemicals, pulp and paper, mineral processing, power and waste water treatment. Control performance assessment and monitoring applications have become mainstream in these industries and are changing the maintenance methodology surrounding control assets from predictive to condition based. The large numbers of these assets on most sites compared to the number of maintenance and control personnel have made monitoring and diagnosing control problems challenging. For this reason, automated controller performance monitoring technologies have been readily embraced by these industries.

This entry discusses the theory as well as practical application of controller performance monitoring tools as a requisite for monitoring and maintaining basic as well as advanced process control (APC) assets in the process industry. The section begins with the introduction to the theory of performance assessment as a technique for assessing the performance of the basic control loops in a plant. Performance assessment allows detection of performance degradation in the basic control loops in a plant by monitoring the variance in the process variable and comparing it to that of a minimum variance controller. Other metrics of controller performance are also reviewed. The resulting indices of performance give an indication of the level of performance of the controller and an indication of the action required to improve its performance; the diagnosis of poor performance may lead one to look at remediation alternatives such as: retuning controller parameters or process reengineering to reduce delays or implementation of feed-forward control or attribute poor performance to faulty actuators or other process nonlinearities.

Keywords

Time series analysis; Minimum variance control; Control loop performance assessment; Performance monitoring; Fault detection and diagnosis

Introduction

A typical industrial process, as in a petroleum refinery or a petrochemical complex, includes thousands of control loops. Instrumentation technicians and engineers maintain and service these loops, but rather infrequently. However, industrial studies have shown that as many as 60 % of control loops may have poor tuning or configuration or actuator problems and may therefore be responsible for suboptimal process performance. As a result, monitoring of such control strategies to detect and diagnose cause(s) of unsatisfactory performance has received increasing attention from industrial engineers. Specifically the methodology of data-based controller performance monitoring (CPM) is able to answer questions such as the following: Is the controller doing its job satisfactorily and if not, what is the cause of poor performance?

The performance of process control assets is monitored on a daily basis and compared with industry benchmarks. The monitoring system also provides diagnostic guidance for poorly performing control assets. Many industrial sites have

established reporting and remediation workflows to ensure that improvement activities are carried out in an expedient manner. Plant-wide performance metrics can provide insight into company-wide process control performance. Closed-loop tuning and modeling tools can also be deployed to aid with the improvement activities. Survey articles by Thornhill and Horch (2007) and Shardt et al. (2012) provide a good overview of the overall state of CPM and the related diagnosis issues. CPM software is now readily available from most DCS vendors and has already been implemented successfully at many large-scale industrial sites throughout the world.

Univariate Control Loop Performance Assessment with Minimum Variance Control as Benchmark

It has been shown by Harris (1989) that for a system with time delay d, a portion of the output variance is feedback control invariant and can be estimated from routine operating data. This is the so-called minimum variance output. Consider the closed-loop system shown in Fig. 1, where Q is the controller transfer function, $\tilde{T}$ is the process transfer function, d is the process delay (in terms of sample periods), and N is the disturbance transfer function driven by random white-noise sequence, a_t.

In the regulatory mode (when the set point is constant), the closed-loop transfer function relating the process output and the disturbance is given by

Closed-loop response: $y_t = \left(\dfrac{N}{1 + q^{-d}\tilde{T}Q} \right) a_t$
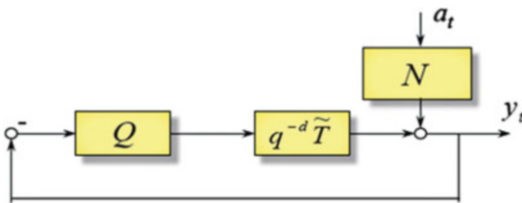
Controller Performance Monitoring, Fig. 1 Block diagram of a regulatory control loop

Note that all transfer functions are expressed for the discrete time case in terms of the backshift operator, q^{-1}. N represents the disturbance transfer function with numerator and denominator polynomials in q^{-1}. The division of the numerator by the denominator can be rewritten as: $N = F + q^{-d}R$, where the quotient term, $F = F_0 + F_1 q^{-1} + \cdots + F_{d-1} q^{-(d-1)}$ is a polynomial of order $(d-1)$ and the remainder, R is a transfer function. The closed-loop transfer function can be reexpressed, after algebraic manipulation as

$$y_t = \left(\frac{N}{1 + q^{-d}\tilde{T}Q} \right) a_t$$

$$= \left(\frac{F + q^{-d}R}{1 + q^{-d}\tilde{T}Q} \right) a_t$$

$$= \left(\frac{F\left(1 + q^{-d}\tilde{T}Q\right) + q^{-d}\left(R - F\tilde{T}Q\right)}{1 + q^{-d}\tilde{T}Q} \right) a_t$$

$$= \left(F + q^{-d}\frac{R - F\tilde{T}Q}{1 + q^{-d}\tilde{T}Q} \right) a_t$$

$$= \underbrace{F_0 a_t + F_1 a_{t-1} + \cdots + F_{d-1} a_{t-d+1}}_{e_t}$$

$$+ \underbrace{L_0 a_{t-d} + L_1 a_{t-d-1} + \cdots}_{w_{t-d}}$$

The closed-loop output can then be expressed as

$$y_t = e_t + w_{t-d}$$

where $e_t = F a_t$ corresponds to the first $d - 1$ lags of the closed-loop expression for the output, y_t, and more importantly is independent of the controller, Q, or it is controller invariant, while w_{t-d} is dependent on the controller. The variance of the output is then given by

$$Var(y_t) = Var(e_t) + Var(w_{t-d}) \geq Var(e_t)$$

Since e_t is controller invariant, it provides the lower bound on the output variance. This is naturally achieved if $w_{t-d} = 0$, that is, when $R = F\tilde{T}Q$ or when the controller is a minimum variance controller with $Q = \frac{R}{F\tilde{T}}$. If the total output variance is denoted as $Var(y_t) = \sigma^2$, then

the lowest achievable variance is $Var(e_t) = \sigma_{mv}^2$. To obtain an estimate of the lowest achievable variance from the time series of the process output, one needs to model the closed-loop output data y_t by a moving average process such as

$$y_t = \underbrace{f_0 a_t + f_1 a_{t-1} + \cdots + f_{d-1} a_{t-(d-1)}}_{e_t}$$
$$+ f_d a_{t-d} + f_{d+1} a_{t-(d+1)} + \cdots \quad (1)$$

The controller-invariant term e_t can then be estimated by time series analysis of routine closed-loop operating data and subsequently used as a benchmark measure of theoretically achievable absolute lower bound of output variance to assess control loop performance. Harris (1989),

Desborough and Harris (1992), and Huang and Shah (1999) have derived algorithms for the calculation of this minimum variance term.

Multiplying Eq. (1) by $a_t, a_{t-1}, \ldots, a_{t-d+1}$, respectively, and then taking the expectation of both sides of the equation yield the sample covariance terms:

$$\left. \begin{array}{c} r_{ya}(0) = E[y_t a_t] = f_0 \sigma_a^2 \\ r_{ya}(1) = E[y_t a_{t-1}] = f_1 \sigma_a^2 \\ r_{ya}(2) = E[y_t a_{t-2}] = f_2 \sigma_a^2 \\ \vdots \\ r_{ya}(d-1) = E[y_t a_{t-d+1}] = f_{d-1} \sigma_a^2 \end{array} \right\} \quad (2)$$

The minimum variance or the invariant portion of output variance is

$$\left. \begin{array}{l} \sigma_{mv}^2 = \left(f_0^2 + f_1^2 + f_2^2 + \cdots + f_{d-1}^2 \right) \sigma_a^2 \\ \\ = \left[\left(\dfrac{r_{ya}(0)}{\sigma_a^2} \right)^2 + \left(\dfrac{r_{ya}(1)}{\sigma_a^2} \right)^2 + \left(\dfrac{r_{ya}(2)}{\sigma_a^2} \right)^2 + \left(\dfrac{r_{ya}(d-1)}{\sigma_a^2} \right)^2 \right] \sigma_a^2 \\ \\ = \left[r_{ya}^2(0) + r_{ya}^2(1) + r_{ya}^2(2) + \cdots + r_{ya}^2(d-1) \right] / \sigma_a^2 \end{array} \right\} \quad (3)$$

A measure of controller performance index can then be defined as

$$\eta(d) = \sigma_{mv}^2 / \sigma_y^2 \quad (4)$$

Substituting Eq. (3) into Eq. (4) yields

$$\eta(d) = \left[r_{ya}^2(0) + r_{ya}^2(1) + r_{ya}^2(2) + \cdots + r_{ya}^2(d-1) \right] / \sigma_y^2 \sigma_a^2$$
$$= \rho_{ya}^2(0) + \rho_{ya}^2(1) + \rho_{ya}^2(2) + \cdots + \rho_{ya}^2(d-1)$$
$$= ZZ^T$$

where Z is the vector of cross correlation coefficients between y_t and a_t for lags 0 to $d-1$ and is denoted as

$$Z = \left[\rho_{ya}(0) \, \rho_{ya}(1) \, \rho_{ya}(2) \ldots \rho_{ya}(d-1) \right]$$

Although a_t is unknown, it can be replaced by the estimated innovation sequence $\hat{a}_t$. The estimate $\hat{a}_t$ is obtained by whitening the process output variable y_t via time series analysis. This algorithm is denoted as the FCOR algorithm

for Filtering and CORrelation analysis (Huang and Shah 1999). This derivation assumes that the delay, d, be known a priori. In practice, however, a priori knowledge of time delays may not always be available. It is therefore useful to assume a range of time delays and then calculate performance indices over this range of the time delays. The indices over a range of time delays are also known as extended horizon performance indices (Thornhill et al. 1999). Through pattern recognition, one can tell the performance of the loop by visualizing the patterns of the performance indices versus time delays. There is a clear relation between performance indices curve and the impulse response curve of the control loop.

Consider a simple case where the process is subject to random disturbances. Figure 2 is one example of performance evaluation for a control loop in the presence of disturbances. This figure shows time-series of process variable data for both loops in the left column, closed-loop impulse responses (middle column) and corresponding performance indices (labeled as PI on the right column). From the impulse responses, one can see that the loop under the first set of tuning constants (denoted as TAG1.PV) has better performance; the loop under the second set of tuning constants (denoted as TAG5.PV) has oscillatory behavior, indicating a relatively poor control performance. With performance index "1" indicating the best possible performance and index "0" indicating the worst performance, performance indices for the first controller tuning (shown on the upper-right plot) approach "1" within 4 time lags, while performance indices for the second controller tuning (shown on the bottom-right plot) take 10 time lags to approach "0.7." In addition, performance indices for the second tuning show ripples as they approach an asymptotic limit, indicating a possible oscillation in the loop.

Notice that one cannot rank performance of these two controller settings from the noisy time-series data. Instead, we can calculate performance indices over a range of time delays (from 1 to 10). The result is shown on the right column plots of Fig. 2. These simulations correspond to the same process with different controller tuning constants.

It is clear from these plots that performance indices trajectory depends on dynamics of the disturbance and controller tuning.

It is important to note that the minimum variance is just one of several benchmarks for obtaining a controller performance metric. It is seldom practical to implement minimum variance control as it typically will require aggressive actuator action. However, the minimum variance benchmark serves to provide an indication of the opportunity in improving control performance; that is, should the performance index $\eta(d)$ be near or just above zero, then it gives the user an idea of the benefits possible in improving the control performance of that loop.

Performance Assessment and Diagnosis of Univariate Control Loop Using Alternative Performance Indicators

In addition to the performance index for performance assessment, there are several alternative indicators of control loop performance. These are discussed next.

Autocorrelation function: The autocorrelation function (ACF) of the output error, shown in Fig. 3, is an approximate measure of how close the existing controller is to minimum variance condition or how predictable the error is over the time horizon of interest. If the controller is under minimum variance condition then the autocorrelation function should decay to zero after "$d - 1$" lags where "d" is the delay of the process. In other words, there should be no predictable information beyond time lag $d - 1$. The rate at which the autocorrelation decays to zero after "$d - 1$" lags indicates how close the existing controller is to the minimum variance condition. Since it is straightforward to calculate autocorrelation using process data, the autocorrelation function is often used as a first-pass test before carrying out further performance analysis.
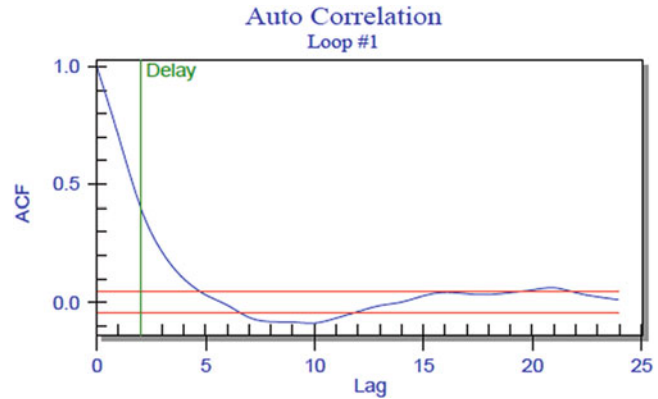
Impulse response: An impulse response function curve represents the closed-loop impulse
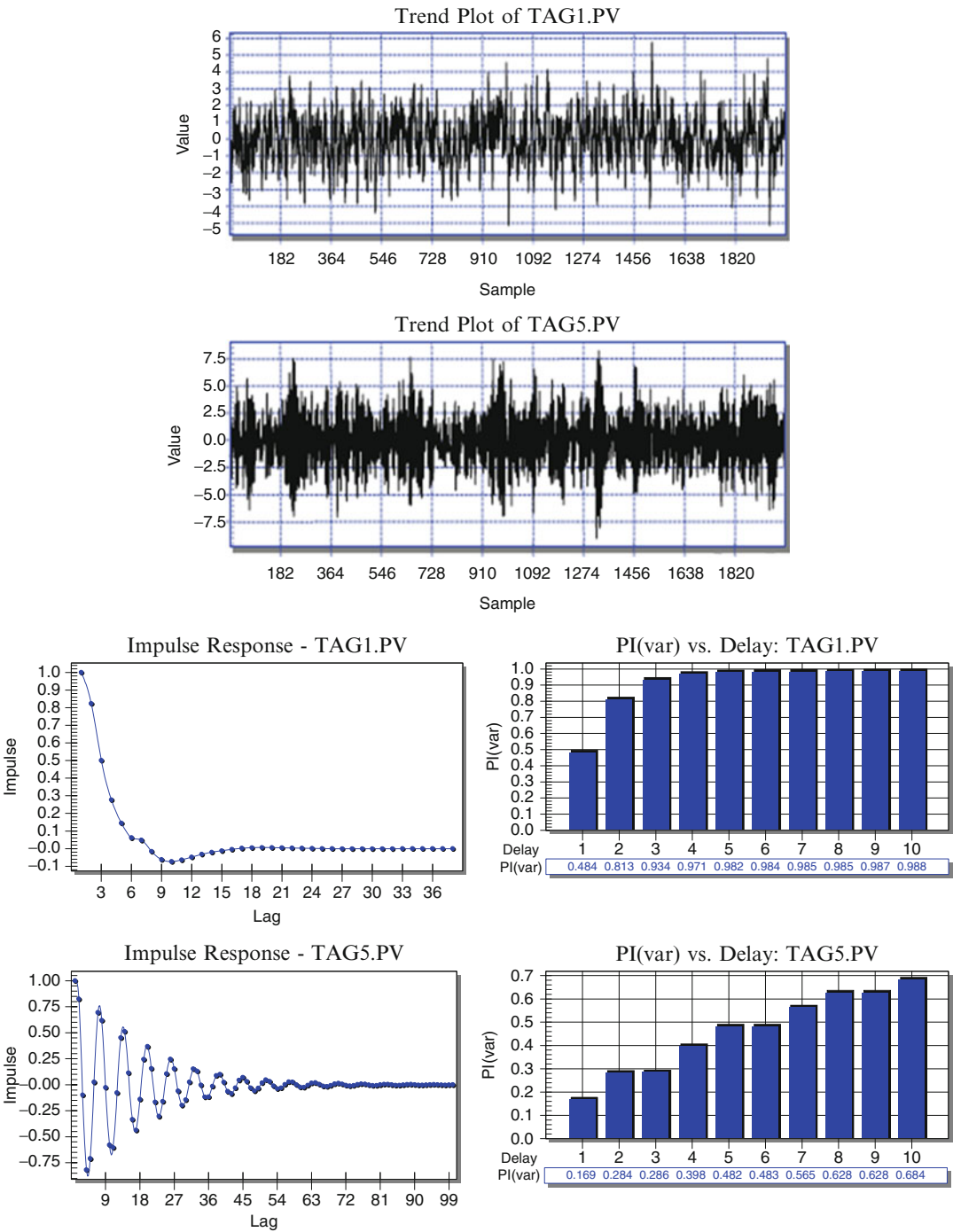
Controller Performance Monitoring, Fig. 2 Time series of process variable (top), corresponding impulse responses (left column) and their performance indices (right column).
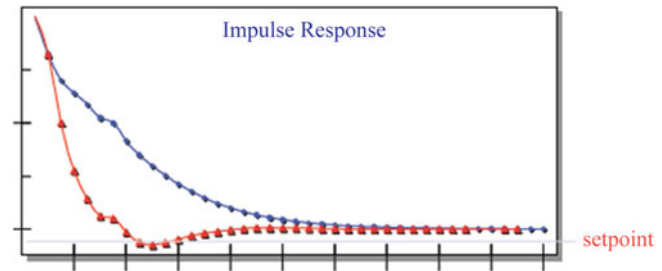
Controller Performance Monitoring, Fig. 3
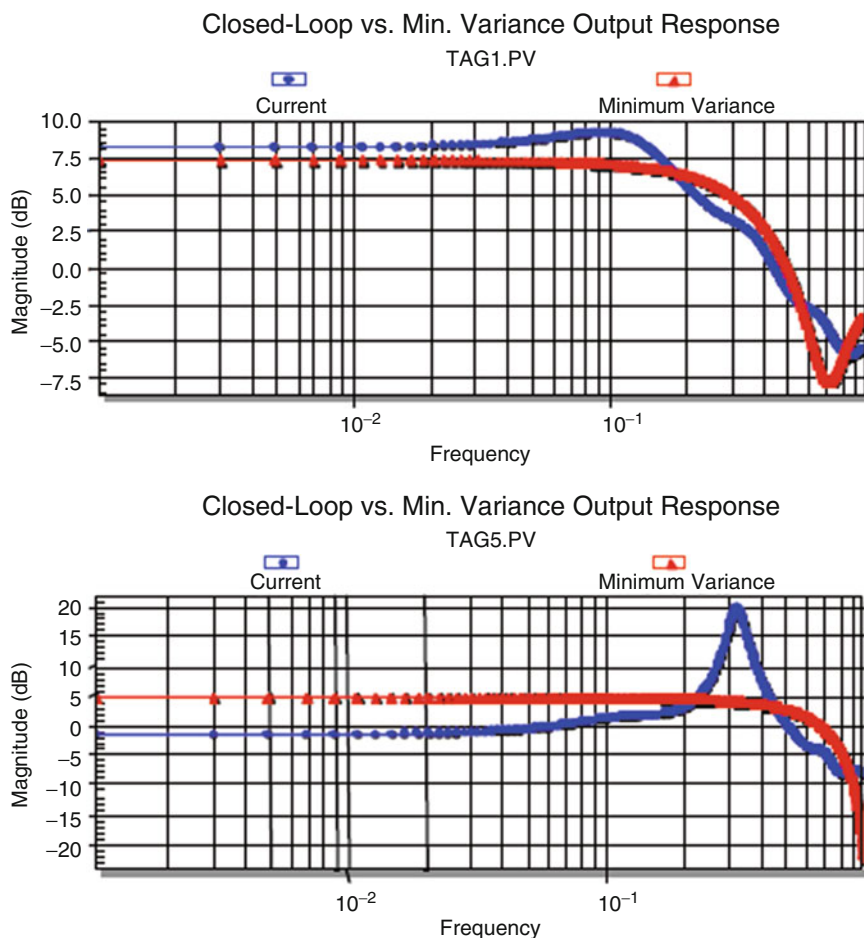Autocorrelation function of the controller error

Controller Performance Monitoring, Fig. 4
Impulse responses estimated from routine operating data

response between the whitened disturbance sequence and the process output. This function is a direct measure of how well the controller is performing in rejecting disturbances or tracking set-point changes. Under stochastic framework, this impulse response function may be calculated using time series model such as an Autoregressive Moving Average (ARMA) or Autoregressive with Integrated Moving Average (ARIMA) model. Once an ARMA type of time series model is estimated, the infinite-order moving average representation of the model shown in Eq. (1) can be obtained through a long division of the ARMA model. As shown in Huang and Shah (1999), the coefficients of the moving average model, Eq. (1), are the closed-loop impulse response coefficients of the process between whitened disturbances and the process output. Figure 4 shows closed-loop impulse responses of a control loop with two different control tunings. Clearly they denote two different closed-loop dynamic responses: one is slow and smooth, and the other one is relatively fast and slightly oscillatory. The sum of square of the impulse response coefficients is the variance of the data.

Spectral analysis: The closed-loop frequency response of the process is an alternative way to assess control loop performance. Spectral analysis of output data easily allows one to detect oscillations, offsets, and measurement noise present in the process. The closed-loop frequency response is often plotted together with the closed-loop frequency response under minimum variance control. This is to check the possibility of performance improvement through controller tunings. The comparison gives a measure of how close the existing controller is to the minimum variance condition. In addition, it also provides the frequency range in which the controller significantly deviates from minimum variance condition. Large deviation in the low-frequency range typically indicates lack of integral action or weak proportional gain. Large peaks in the medium-frequency range typically indicate an overtuned controller or presence of oscillatory disturbances. Large deviation in the high-frequency range typically indicates significant measurement noise. As an illustrative example, frequency responses of two control loops are shown in Fig. 5. The left graph of the figure shows that closed-loop

Closed-Loop vs. Min. Variance Output Response
TAG1.PV

Closed-Loop vs. Min. Variance Output Response
TAG5.PV

Controller Performance Monitoring, Fig. 5   Frequency response estimated from routine operating data

frequency response of the existing controller is almost the same as the frequency response under minimum variance control. A peak at the mid-frequency indicates possible overtuned control. The right graph of Fig. 5 shows that the frequency response of the existing controller is oscillatory, indicating a possible overtuned controller or the presence of an oscillatory disturbance at the peak frequency; otherwise the controller is close to minimum variance condition.

Segmentation of performance indices: Most process data exhibit time- varying dynamics; i.e., the process transfer function or the disturbance transfer function is time variant. Performance assessment with a non-overlapping sliding data window that can track time-varying dynamics

is therefore often desirable. For example, segmentation of data may lead to some insight into any cyclical behavior of the process variation in controller performance during, e.g., day/night or due to shift change. Figure 6 is an example of performance segmentation over a 200 data point window.

Performance Assessment of Univariate Control Loops Using User-Specified Benchmarks

The increasing level of global competitiveness has pushed chemical plants into high-performance operating regions that require advanced process control technology. See the

articles ► Control Hierarchy of Large Processing Plants: An Overview and ► Control Structure Selection. Consequently, the industry has an increasing need to upgrade the conventional PID controllers to advanced control systems. The most natural questions to ask for such an upgrading are as follows. Has the advanced controller improved performance as expected? If yes, where is the improvement and can it be justified? Has the advanced controller been tuned to its full capacity? Can this improvement also be achieved by simply retuning the existing traditional (e.g., PID) controllers? (see ► PID Control). In other words, what is the cost versus benefit of implementing an advanced controller? Unlike performance assessment using minimum variance control as benchmark, the solution to this problem does not require a priori knowledge of time delays. Two possible relative benchmarks may be chosen: one is the historical data benchmark or reference data set benchmark, and the other is a user-specified benchmark.

The purpose of reference data set benchmarking is to compare performance of the existing controller with the previous controller during the "normal" operation of the process. This reference data set may represent the process when the controller performance is considered satisfactory with respect to meeting the performance objectives. The reference data set should be representative of the normal conditions that the process is expected to operate at; i.e., the disturbances and set-point changes entering into the process should not be unusually different. This analysis provides the user with a relative performance index (RPI) which compares the existing control loop performance with a reference control loop benchmark chosen by the user. The RPI is bounded by $0 \leq RPI \leq \infty$, with "<1" indicating deteriorated performance, "1" indicating no change of performance, and ">1" indicating improved performance. Figure 6 shows a result of reference data set benchmarking. The impulse response of the benchmark or reference data smoothly decays to zero, indicating good performance of the controller. After one increases the proportional gain of the controller, the impulse response

shows oscillatory behavior, with an RPI = 0.4, indicating deteriorated performance due to the oscillation.

In some cases one may wish to specify certain desired closed-loop dynamics and carry out performance analysis with respect to such desired dynamics. One such desired dynamic benchmark is the closed-loop settling time. As an illustrative example, Fig. 8 shows a system where a settling time of ten sampling units is desired for a process with a delay of five sampling units. The impulse responses show that the existing loop is close to the desired performance, and the value of RPI = 0.9918 confirms this. Thus no further tuning of the loop is necessary.
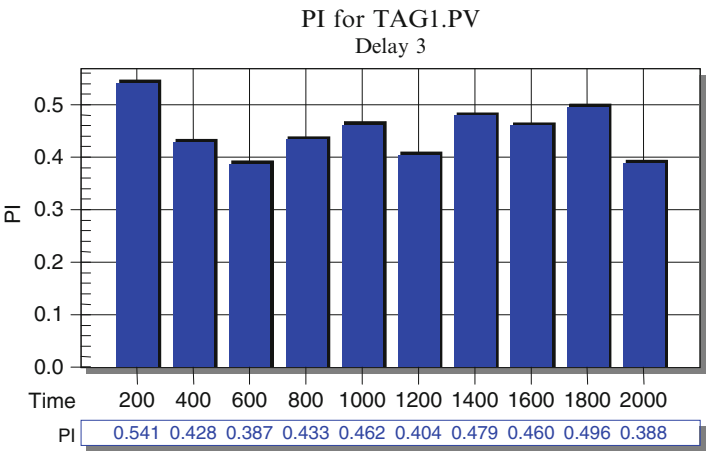
Diagnosis of Poorly Performing Loops

Whereas detection of poorly performing loops is now relatively simple, the task of diagnosing reason(s) for poor performance and how to "mend" the loop is generally not straightforward. The reasons for poor performance could be any one of interactions between various control loops, overtuned or undertuned controller settings, process nonlinearity, poor controller configuration (meaning the choice of pairing a process (or controlled) variable with a manipulative variable loop), or actuator problems such as stiction, large delays, and severe disturbances. Several studies have focused on the diagnosis issues related to actuator problems (Håagglund 2002; Choudhury et al. 2008; Srinivasan and Rengaswamy 2008; Xiang and Lakshminarayanan 2009; de Souza et al. 2012). Shardt et al. (2012) has given an overview of the overall state of CPM and the related diagnosis issues.

Industrial Applications of CPM Technology

As remarked earlier, CPM software is now readily available from most DCS vendors and has already been implemented successfully at several large-scale industrial sites. A summary of just
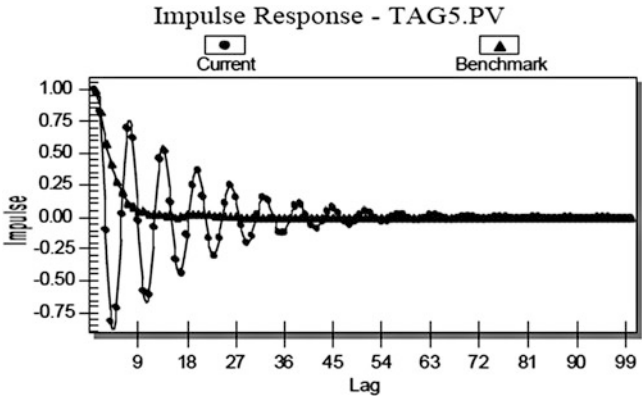
Controller Performance Monitoring, Fig. 6 Performance indices for segmented data (each of window length 200 points)
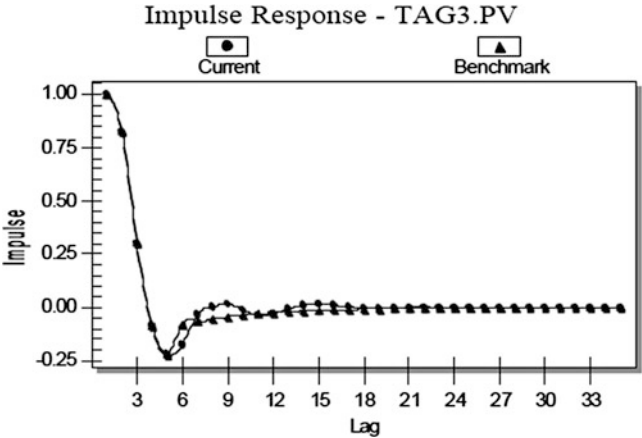

PI for TAG1.PV
Delay 3

Time	200	400	600	800	1000	1200	1400	1600	1800	2000
PI	0.541	0.428	0.387	0.433	0.462	0.404	0.479	0.460	0.496	0.388

Controller Performance Monitoring, Fig. 7 Reference benchmarking based on impulse responses

Impulse Response - TAG5.PV

Controller Performance Monitoring, Fig. 8 User-specified benchmark based on impulse responses

Impulse Response - TAG3.PV

two of many large-scale industrial implementations of CPM technology appears below. It gives a clear evidence of the impact of this control technology and how readily it has been embraced by industry (Shah et al. 2014).

BASF Controller Performance Monitoring Application

As part of its excellence initiative OPAL 21 (Optimization of Production Antwerp and Ludwigshafen), BASF has implemented the CPM strategy on more than 30,000 control loops at its Ludwigshafen site in Germany and on over 10,000 loops at its Antwerp production facility in Belgium. The key factor in using this technology effectively is to combine process knowledge, basic chemical engineering, and control expertise to develop solutions for the indicated control problems that are diagnosed in the CPM software (Wolff et al. 2012).

Saudi Aramco Controller Performance Monitoring Practice

As part of its process control improvement initiative, Saudi Aramco has deployed CPM on approximately 15,000 PID loops, 50 MPC applications, and 500 smart positioners across multiple operating facilities.

The operational philosophy of the CPM engine is incorporated in the continuous improvement process at BASF and Aramco, whereby all loops are monitored in real-time and a holistic performance picture is obtained for the entire plant. Unit-wide performance metrics are displayed in effective color-coded graphic forms to effectively convey the analytics information of the process.

Concluding Remarks

In summary, industrial control systems are designed and implemented or upgraded with a particular objective in mind. The controller performance monitoring methodology discussed here will permit automated and repeated reviews of the design, tuning, and upgrading of the control loops. Poor design, tuning, or upgrading of the control loops can be detected, and repeated performance monitoring will indicate which loops should be retuned or which loops have not been effectively upgraded when changes in the disturbances, in the process, or in the controller itself occur. Obviously better design, tuning, and upgrading will mean that the process will operate at a point close to the economic optimum, leading to energy savings, improved safety, efficient utilization of raw materials, higher product yields, and more consistent product qualities. This entry has summarized the major features available in recent commercial software packages for control loop performance assessment. The illustrative examples have demonstrated the applicability of this new technique when applied to process data.

This entry has also illustrated how controllers, whether in hardware or software form, should be treated like "capital assets" and how there should be routine monitoring to ensure that they perform close to the economic optimum and that the benefits of good regulatory control will be achieved.

Cross-References

▶ Control Hierarchy of Large Processing Plants: An Overview
▶ Control Structure Selection
▶ Fault Detection and Diagnosis
▶ PID Control
▶ Statistical Process Control in Manufacturing

Bibliography

Choudhury MAAS, Shah SL, Thornhill NF (2008) Diagnosis of process nonlinearities and valve stiction: data driven approaches. Springer-Verlag, Sept. 2008, ISBN:978-3-540-79223-9

Desborough L, Harris T (1992) Performance assessment measure for univariate feedback control. Can J Chem Eng 70:1186–1197

Desborough L, Miller R (2002) Increasing customer value of industrial control performance monitoring: Honeywell's experience. In: AIChE symposium series. American Institute of Chemical Engineers, New York, pp 169–189; 1998

de Prada C (2014) Overview: control hierarchy of large processing plants. In: Encyclopedia of Systems and Control. Springer, London

de Souza LCMA, Munaro CJ, Munareto S (2012) Novel model-free approach for stiction compensation in control valves. Ind Eng Chem Res 51(25):8465–8476

Håagglund T (2002) A friction compensator for pneumatic control valves. J Process Control 12(8):897–904

Harris T (1989) Assessment of closed loop performance. Can J Chem Eng 67:856–861

Huang B, Shah SL (1999) Performance assessment of control loops: theory and applications. Springer-Verlag, October 1999, ISBN: 1-85233-639-0.

Shah SL, Nohr M, Patwardhan R (2014) Success stories in control: controller performance monitoring. In: Samad T, Annaswamy AM (eds) The impact of control technology, 2nd edn. www.ieeecss.org

Shardt YAW, Zhao Y, Lee KH, Yu X, Huang B, Shah SL (2012) Determining the state of a process control system: current trends and future challenges. Can J Chem Eng 90(2):217–245

Srinivasan R, Rengaswamy R (2008) Approaches for efficient stiction compensation in process control valves. Comput Chem Eng 32(1):218–229

Skogestad S (2014) Control structure selection and plantwide control. In: Encylopedia of Systems and Control. Springer, London

Thornhill NF, Horch A (2007) Advances and new directions in plant-wide controller performance assessment. Control Eng Pract 15(10):1196–1206

Thornhill NF, Oettinger M, Fedenczuk P (1999) Refinery-wide control loop performance assessment. J Process Control 9(2):109–124

Wolff F, Roth M, Nohr A, Kahrs O (2012) Software based control-optimization for the chemical industry. VDI, Tagungsband "Automation 2012", 13/14.06 2012, Baden-Baden

Xiang LZ, Lakshminarayanan S (2009) A new unified approach to valve stiction quantification and compensation. Ind Eng Chem Res 48(7):3474–3483

Cooperative Manipulators

Fabrizio Caccavale
School of Engineering, Università degli Studi della Basilicata, Potenza, Italy

Abstract

This chapter presents an overview of the main issues related to modeling and control of cooperative robotic manipulators. A historical path is followed to present the main research results on cooperative manipulation. Kinematics and dynamics of robotic arms cooperatively manipulating a tightly grasped rigid object are briefly discussed. Then, this entry presents the main strategies for force/motion control of the cooperative system.

Keywords

Introduction

Since the early 1970s, it has been recognized that many tasks, which are difficult or even impossible to execute by a single robotic manipulator, become feasible when two or more manipulators work in a cooperative way. Examples of typical cooperative tasks are the manipulation of heavy and/or large payloads, assembly of multiple parts, and handling of flexible and articulated objects (Fig. 1).

In the 1980s, research achieved several theoretical results related to modeling and control of to single-arm robots; this further fostered research on multi-arm robotic systems. Dynamics

Cooperative Manipulators, Fig. 1 An example of a cooperative robotic work cell composed by two industrial robot arms

and control as well as force control issues have been widely explored along the decade.

In the 1990s, parameterization of the constraint forces/moments acting on the object has been recognized as a key to solving control problems and has been studied in several papers (e.g., Sang et al. 1995; Uchiyama and Dauchez 1993; Walker et al. 1991; Williams and Khatib 1993). Several control schemes for cooperative manipulators based on the sought parameterizations have been designed, including force/motion control (Wen and Kreutz-Delgado 1992) and impedance control (Bonitz and Hsia 1996; Schneider and Cannon 1992). Other approaches are adaptive control (Hu et al. 1995), kinematic control (Chiacchio et al. 1996), task-space regulation (Caccavale et al. 2000), and model-based coordinated control (Hsu 1993). Other important topics investigated in the 1990s were the definition of user-oriented task-space variables for coordinated control (Caccavale et al. 2000; Chiacchio et al. 1996), the development of meaningful performance measures (Chiacchio et al. 1991a,b) for multi-arm systems, and the problem of load sharing (Walker et al. 1989).

Most of the abovementioned works assume that the cooperatively manipulated object is rigid and tightly grasped. However, since the 1990s, several research efforts have been focused on the control of cooperative flexible manipulators (Yamano et al. 2004), since flexible-arm robot merits (lightweight structure, intrinsic compliance, and hence safety) can be conveniently exploited in cooperative manipulation. Other research efforts have been focused on the control of cooperative systems for the manipulation of flexible objects (Yukawa et al. 1996) as well.

Modeling, Load Sharing, and Performance Evaluation

The first modeling goal is the definition of suitable variables describing the kinetostatics of a cooperative system. Hereafter, the main results available are summarized for a dual-arm system

composed by two cooperative manipulators grasping a common object.

The kinetostatic formulation proposed by Uchiyama and Dauchez (1993), i.e., the so-called *symmetric formulation*, is based on kinematic and static relationships between generalized forces/velocities acting at the object and their counterparts acting at the manipulators end effectors. To this aim, the concept of *virtual stick* is defined as the vector which determines the position of an object-fixed coordinate frame with respect to the frame attached to each robot end effector (Fig. 2). When the object grasped by the two manipulators can be considered rigid and tightly attached to each end effector, then the virtual stick behaves as a rigid stick fixed to each end effector.

According to the symmetric formulation, the vector, h, collecting the generalized forces (i.e., forces and moments) acting at each end effector is given by

$$h = W^\dagger h_E + V h_I, \qquad (1)$$

where W is the so-called *grasp matrix*, the columns of V span the null space of the
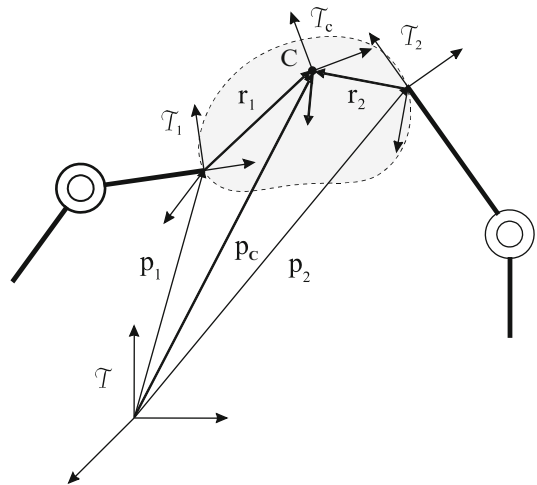
Cooperative Manipulators, Fig. 2 Grasp geometry for a two-manipulator cooperative system manipulating a common object. The vectors r_1 and r_2 are the *virtual sticks*, $\mathcal{T}_c$ is the coordinate frame attached to the object, and $\mathcal{T}_1$ and $\mathcal{T}_2$ are the coordinate frames attached to each end effector

grasp matrix, and h_I is the generalized force vector which does not contribute to the object's motion, i.e., it represents internal loading of the object (mechanical stresses) and is termed as *internal forces*, while h_E represents the vector of *external forces*, i.e., forces and moments causing the object's motion. Later, a *task-oriented formulation* has been proposed (Chiacchio et al. 1996), aimed at defining a *cooperative task space* in terms of *absolute* and *relative* motion of the cooperative system, which can be directly computed from the position and orientation of the end-effector coordinate frames.

The dynamics of a cooperative multi-arm system can be written as the dynamics of the single manipulators together with the closed-chain constraints imposed by the grasped object. By eliminating the constraints, a reduced-order model can be obtained (Koivo and Unseren 1991).

Strongly related to kinetostatics and dynamics of cooperative manipulators is the load sharing problem, i.e., distributing the load among the arms composing the system, which has been solved, e.g., in Walker et al. (1989). A very relevant problem related to the load sharing is that of robust holding, i.e., the problem of determining forces/moments applied to object by the arms, in order to keep the grasp even in the presence of disturbing forces/moments.

A major issue in robotic manipulation is the performance evaluation via suitably defined indexes (e.g., manipulability ellipsoids). These concepts have been extended to multi-arm robotic systems in Chiacchio et al. (1991a,b). Namely, by exploiting the kinetostatic formulations described above, velocity and force manipulability ellipsoids can be defined, by regarding the whole cooperative system as a mechanical transformer from the joint space to the cooperative task space. The manipulability ellipsoids can be seen as performance measures aimed at determining the attitude of the system to cooperate in a given configuration.

Finally, it is worth mentioning the strict relationship between problems related to grasping of objects by fingers/hands and those related to cooperative manipulation. In fact, in both cases, multiple manipulation structures grasp a commonly manipulated object. In multifingered hands, only some motion components are transmitted through the contact point to the manipulated object (unilateral constraints), while cooperative manipulation via robotic arms is achieved by rigid (or near-rigid) grasp points and interaction takes place by transmitting all the motion components through the grasping points (bilateral constraints). While many common problems between the two fields can be tackled in a conceptually similar way (e.g., kinetostatic modeling, force control), many others are specific of each of the two application fields (e.g., form and force closure for multifingered hands).

Control

When a cooperative multi-arm system is employed for the manipulation of a common object, it is important to control both the absolute motion of the object and the internal stresses applied to it. Hence, most of the control approaches to cooperative robotic systems can be classified as force/motion control schemes.

Early approaches to the control of cooperative systems were based on the *master/slave* concept. Namely, the cooperative system is decomposed in a position-controlled master arm, in charge of imposing the absolute motion of the object, and the force-controlled slave arms, which are to follow (as smoothly as possible) the motion imposed by the master. A natural evolution of the above-described concept has been the so-called *leader/follower* approach, where the follower arm reference motion is computed via closed-chain constraints. However, such approaches suffered from implementation issues, mainly due to the fact that the compliance of the slave arms has to be very large, so as to smoothly follow the motion imposed by the master arm. Moreover, the roles of the master and slave (leader and follower) may need to be changed during the task execution.

Due to the abovementioned limitations, more natural nonmaster/slave approaches have been pursued later, where the cooperative system is seen as a whole. Namely, the reference motion

of the object is used to determine the motion of all the arms in the system and the interaction forces are measured and fed back so as to be directly controlled. To this aim, the mappings between forces and velocities at the end effector of each manipulator and their counterparts at the manipulated object are considered in the design of the control laws.

An approach, based on the classical hybrid force/position control scheme, has been proposed in Uchiyama and Dauchez (1993), by exploiting the symmetric formulation described in the previous section.

In Wen and Kreutz-Delgado (1992) a Lyapunov-based approach is pursued to devise force/position PD-type control laws. This approach has been extended in Caccavale et al. (2000), where kinetostatic filtering of the control action is performed, so as to eliminate all the components of the control input which contribute to internal stresses at the object.

A further improvement of the PD plus gravity compensation control approach has been achieved by introducing a full model compensation, so as to achieve feedback linearization of the closed-loop system. The feedback linearization approach formulated at the operational space level is the base of the so-called *augmented object* approach (Sang et al. 1995). In this approach, the system is modeled in the operational space as a whole, by suitably expressing its inertial properties via a single augmented inertia matrix M_O, i.e.,

$$M_O(x_E)\ddot{x}_E + c_O(x_E, \dot{x}_E) + g_O(x_E) = h_E, \quad (2)$$

where M_O, c_O, and g_O are the operational space terms modeling, respectively, the inertial properties of the whole system (manipulators and object), the Coriolis/centrifugal/friction terms, and the gravity terms, while x_E is the operational space vector describing the position and orientation of the coordinate frame attached to the grasped object. In the framework of feedback linearization (formulated in the operational space), the problem of controlling the internal forces can be solved, e.g., by resorting to the *virtual linkage*

model (Williams and Khatib 1993) or according to the scheme proposed in Hsu (1993).

An alternative control approach is based on the well-known impedance concept (Bonitz and Hsia 1996; Schneider and Cannon 1992). In fact, when a manipulation system interacts with an external environment and/or other manipulators, large values of the contact forces and moments can be avoided by enforcing a compliant behavior with suitable dynamic features. In detail, the following mechanical impedance behavior between the object displacements and the forces due to the object-environment interaction can be enforced (*external impedance*):

$$M_E \tilde{a}_E + D_E \tilde{v}_E + K_e e_E = h_{\text{env}}, \quad (3)$$

where e_E represents the vector of displacements between object's desired and actual pose, $\tilde{v}_E$ is the difference between the object's desired and actual generalized velocities, $\tilde{a}_E$ is the difference between the object's desired and actual generalized accelerations, and h_{env} is the generalized force acting on the object, due to the interaction with the environment. The impedance dynamics is characterized in terms of given positive definite mass, damping, and stiffness matrices (M_E, D_E, K_E). A mechanical impedance behavior between the ith end-effector displacements and the internal forces can be imposed as well (*internal impedance*):

$$M_{I,i} \tilde{a}_i + D_{I,i} \tilde{v}_i + K_{I,i} e_i = h_{I,i}, \quad (4)$$

where e_i is the vector expressing the displacement between the commanded and the actual pose of the ith end effector, $\tilde{v}_i$ is the vector expressing the difference between commanded and actual generalized velocities of the ith end effector, $\tilde{a}_i$ is the vector expressing the difference between commanded and actual generalized accelerations of the ith end effector, and $h_{I,i}$ is the contribution of the ith end effector to the internal force. Again, the impedance dynamics is characterized in terms of given positive definite mass, damping, and stiffness matrices ($M_{I,i}, D_{I,i}, K_{I,i}$). More recently, an impedance scheme for control

of both external forces and internal forces has
been proposed (Caccavale et al. 2008).

Summary and Future Directions

This entry has provided a brief survey of the
main issues related to cooperative robots, with
special emphasis on modeling and control prob-
lems. Among several open research topics in
cooperative manipulation, it is worth mentioning
the problem of cooperative transportation and
manipulation of objects via multiple mobile ma-
nipulators. In fact, although notable results have
been already devised in Khatib et al. (1996),
the foreseen use of robotic teams in industrial
settings (hyperflexible robotic work cells) and/or
in collaboration with humans (robotic coworker
concept) raises new challenges related to auton-
omy and safety of multiple mobile manipulators.
Also, an emerging application field is given by
cooperative systems composed by multiple aerial
vehicle-manipulator systems (see, e.g., Fink et al.
2011).

Cross-References

▶ Force Control in Robotics
▶ Robot Grasp Control
▶ Robot Motion Control

Recommended Reading

An overview of the field of cooperative ma-
nipulation can be found also in Caccavale and
Uchiyama (2008), where a more extended lit-
erature review and further technical details are
provided. Seminal contributions to control of co-
operative manipulators can be found in Chiacchio
et al. (1991a), Koivo and Unseren (1991), Sang
et al. (1995), Uchiyama and Dauchez (1993),
Walker et al. (1989), Wen and Kreutz-Delgado
(1992), and Williams and Khatib (1993).

Bibliography

Bonitz RG, Hsia TC (1996) Internal force-based
    impedance control for cooperating manipulators. IEEE
    Trans Robot Autom 12:78–89
Caccavale F, Uchiyama M (2008) Cooperative manipula-
    tors. In: Siciliano B, Khatib O (eds) Springer handbook
    of robotics – chapter 29. Springer, Heidelberg
Caccavale F, Chiacchio P, Chiaverini S (2000) Task-space
    regulation of cooperative manipulators. Automatica
    36:879–887
Caccavale F, Chiacchio P, Marino A, Villani L (2008)
    Six-DOF impedance control of dual-arm coopera-
    tive manipulators. IEEE/ASME Trans Mechatron 13:
    576–586
Chiacchio P, Chiaverini S, Sciavicco L, Siciliano B
    (1991a) Global task space manipulability ellipsoids
    for multiple arm systems. IEEE Trans Robot Autom
    7:678–685
Chiacchio P, Chiaverini S, Sciavicco L, Siciliano B
    (1991b) Task space dynamic analysis of multiarm
    system configurations. Int J Robot Res 10:708–715
Chiacchio P, Chiaverini S, Siciliano B (1996) Direct and
    inverse kinematics for coordinated motion tasks of
    a two-manipulator system. ASME J Dyn Syst Meas
    Control 118:691–697
Fink J, Michael N, Kim S, Kumar V (2011) Planning and
    control for cooperative manipulation and transporta-
    tion with aerial robots. Int J Robot Res 30:324–334
Hsu P (1993) Coordinated control of multiple manipulator
    systems. IEEE Trans Robot Autom 9:400–410
Hu Y-R, Goldenberg AA, Zhou C (1995) Motion and
    force control of coordinated robots during constrained
    motion tasks. Int J Robot Res 14:351–365
Khatib O, Yokoi K, Chang K, Ruspini D, Holmberg
    R, Casal A (1996) Coordination and decentralized
    cooperation of multiple mobile manipulators. J Robot
    Systems 13:755Ű-764
Koivo AJ, Unseren MA (1991) Reduced order model and
    decoupled control architecture for two manipulators
    holding a rigid object. ASME J Dyn Syst Meas Control
    113:646–654
Sang KS, Holmberg R, Khatib O (1995) The augmented
    object model: cooperative manipulation and parallel
    mechanisms dynamics. In: Proceedings of the 2000
    IEEE international conference on robotics and au-
    tomation, San Francisco, pp 470–475
Schneider SA, Cannon Jr RH (1992) Object impedance
    control for cooperative manipulation: theory and ex-
    perimental results. IEEE Trans Robot Autom 8:383–
    394
Uchiyama M, Dauchez P (1993) Symmetric kinematic
    formulation and non-master/slave coordinated control
    of two-arm robots. Adv Robot 7:361–383
Walker ID, Marcus SI, Freeman RA (1989) Distribution
    of dynamic loads for multiple cooperating robot ma-
    nipulators. J Robot Syst 6:35–47
Walker ID, Freeman RA, Marcus SI (1991) Analysis of
    motion and internal force loading of objects grasped

by multiple cooperating manipulators. Int J Robot Res 10:396–409

Wen JT, Kreutz-Delgado K (1992) Motion and force control of multiple robotic manipulators. Automatica 28:729–743

Williams D, Khatib O (1993) The virtual linkage: a model for internal forces in multi-grasp manipulation. In: Proceedings of the 1993 IEEE international conference on robotics and automation, Atlanta, pp 1025–1030

Yamano M, Kim J-S, Konno A, Uchiyama M (2004) Cooperative control of a 3D dual-flexible-arm robot. J Intell Robot Syst 39:1–15

Yukawa T, Uchiyama M, Nenchev DN, Inooka H (1996) Stability of control system in handling of a flexible object by rigid arm robots. In: Proceedings of the 1996 IEEE international conference on robotics and automation, Minneapolis, pp 2332–2339

Cooperative Solutions to Dynamic Games

Alain Haurie
ORDECSYS and University of Geneva, Switzerland
GERAD-HEC Montréal PQ, Canada

Abstract

This article presents the fundamental elements of the theory of cooperative games in the context of dynamic systems. The concepts of Pareto optimality, Nash bargaining solution, characteristic function, cores, and C-optimality are discussed, and some fundamental results are recalled.

Keywords

Introduction

Solution concepts in game theory are regrouped in two main categories called noncooperative and cooperation solutions, respectively. In the seminal book of von Neumann and Morgen-

stern (1944) this categorization is already made. These authors discuss zero-sum (matrix) games in normal form, where the noncooperative solution concept of saddle-point was defined and characterized, and games in characteristic function form, where solution concepts for games of coalitions were introduced. In this article we present the fundamental solution concepts of the theory of cooperative games in the context of dynamical systems. The article is organized as follows: we first recall the papers, which mark the origin of development of a theory of dynamic games; then we recall the basic concept of Pareto optimality proposed as a cooperative solution concept; we present the scalarization technique and the necessary or sufficient optimality conditions for Pareto optimality in mathematical programming and optimal control settings; we then explore the difficulties encountered when one tried to extend the Nash bargaining solution, characteristic function and cores concept to dynamic games; we show the links that exist with the theory of reachability for perturbed dynamic systems.

The Origins

One may consider that the first introduction of a cooperative game solution concept in systems and control science is due to L.A. Zadeh (1963). Two-player zero-sum dynamic games have been studied by R. Isaacs (1954) in a deterministic continuous time setting and by L. Shapley (1953) in a discrete time stochastic setting. Nonzero-sum and m player differential games were introduced by Y.C. Ho and A.W. Starr (1969) and J.H. Case (1969). For these games cooperative solutions can be looked for to complement the noncooperative Nash equilibrium concept.

Cooperation Solution Concept

In cooperative games one is interested in non-dominated solution. This solution type is related to a concept introduced by the well-known economist V. Pareto (1869) in the context of

welfare economics. Consider a system with decision variables $x \in X \subset \mathbb{R}^n$ and m performance criteria $x \to \psi_j(x) \in \mathbb{R}$, $j = 1, \ldots, m$ that one tries to maximize.

Definition 1 The decision $x^* \in X$ is nondominated or Pareto optimal if the following condition holds:

$$\psi_j(x) \geq \psi_j(x^*) \quad \forall j = 1, \ldots m$$
$$\implies \psi_j(x) = \psi_j(x^*) \quad \forall j = 1, \ldots m.$$

In other words it is impossible to give one criterion j a value greater than $\psi_j(x^*)$ without decreasing the value of another criterion, say ℓ, which then takes a value lower than $\psi_\ell(x^*)$.

This vector-valued optimization framework corresponds to a situation where m players are engaged in a game, described in its normal form, where the strategies of the m players constitute the decision vector x and their respective payoffs are given by the m performance criteria $\psi_j(x)$, $j = 1, \ldots, m$. One assumes that these players jointly take a decision that is cooperatively optimal, in the sense that no player can improve his/her payoff without deteriorating the payoff of at least one other player.

The Scalarization Technique

Let $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ be a given m-vector composed of normalized weights that satisfy $r_j > 0$, $j = 1 \ldots, m$ and $\sum_{j=1,\ldots,m} r_j = 1$.

Lemma 1 *Let $x^* \in X$ be a maximum in X for the scalarized criterion $\Psi(x; \mathbf{r}) = \sum_{j=1}^m r_j \psi_j(x)$. Then x^* is a nondominated solution for the multi-objective problem.*

The proof is very simple. Suppose x^* is dominated, then there exists $x^\circ \in X$ such that $\psi_j(x^\circ) \geq \psi_j(x^*)$, $\forall j = 1, \ldots, m$, and $\psi_i(x^\circ) > \psi_i(x^*)$ for one $i \in \{1, \ldots, m\}$. Since all the r_j are > 0, this yields $\sum_{j=1}^m r_j \psi_j(x^\circ) > \sum_{j=1}^m r_j \psi_j(x^*)$, which contradicts the maximizing property of x^*. This result shows that it will be very easy to find many Pareto optimal solutions by varying a strictly positive weighting

of the criteria. But this procedure will not find all of the nondominated solutions.

Conditions for Pareto Optimality in Mathematical Programming

N.O. Da Cunha and E. Polak (1967b) have obtained the first necessary conditions for multiobjective optimization. The problem they consider is

$$\text{Pareto Opt. } \psi_j(x) \quad j = 1, \ldots m$$
$$\text{s.t.}$$
$$\varphi_k(x) \leq 0 \quad k = 1, \ldots p$$

where the functions $x \in \mathbb{R}^n \mapsto \psi_j(x) \in \mathbb{R}$, $j = 1, \ldots, m$, and $x \mapsto \varphi_k(x) \in \mathbb{R}$, $k = 1, \ldots, p$ are continuously differentiable (C^1) and where we assume that the constraint qualification conditions of mathematical programming hold for this problem too. They proved the following theorem.

Theorem 1 *Let x^* be a Pareto optimal solution of the problem defined above. Then there exists a vector λ of p multipliers λ_k, $k = 1, \ldots, p$, and a vector $\mathbf{r} \neq 0$ of m weights $r_j \geq 0$, such that the following conditions hold*

$$\frac{\partial}{\partial x} \mathcal{L}\left(x^*; \mathbf{r}; \lambda\right) = 0$$
$$\varphi_k(x^*) \leq 0$$
$$\lambda_k \varphi_k(x^*) = 0$$
$$\lambda_k \geq 0,$$

where $\mathcal{L}\left(x^*; \mathbf{r}; \lambda\right)$ is the weighted Lagrangian defined by

$$\mathcal{L}\left(x; \mathbf{r}; \lambda\right) = \sum_{j=1}^m r_j \psi_j(x) + \sum_{k=1}^p \lambda_k \varphi_k(x).$$

So there is a local scalarization principle for Pareto optimality.

Maximum Principle

The extension of Pareto optimality concept to control systems was done by several authors (Basile and Vincent 1970; Bellassali and Jourani 2004; Binmore et al. 1986; Blaquière et al. 1972;

Leitmann et al. 1972; Salukvadze 1971; Vincent and Leitmann 1970; Zadeh 1963), the main result being an extension of the maximum principle of Pontryagin. Let a system be governed by state equations:

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

$$u(t) \in U \tag{2}$$

$$x(0) = x_o \tag{3}$$

$$t \in [0, T] \tag{4}$$

where $x \in \mathbb{R}^n$ is the state variable of the system, $u \in U \subset \mathbb{R}^p$ with U compact is the control variable, and $[0, T]$ is the control horizon. The system is evaluated by m performance criteria of the form

$$\psi_j(x(\cdot), u(\cdot)) = \int_0^T g_j(x(t), u(t)) dt + G_j(x(T)), \tag{5}$$

for $j = 1, \ldots, m$. Under the usual assumptions of control theory, i.e., $f(\cdot, \cdot)$ and $g_j(\cdot, \cdot)$, $j = 1, \ldots, m$, being C^1 in x and continuous in u, $G_j(\cdot)$ being C^1 in x, one can prove the following.

Theorem 2 *Let $\{x^*(t) : t \in [0, T]\}$ be a Pareto optimal trajectory, generated at initial state x° by the Pareto optimal control $\{u^*(t) : t \in [0, T]\}$. Then there exist costate vectors $\{\lambda^*(t) : t \in [0, T]\}$ and a vector of positive weights $\mathbf{r} \neq 0 \in \mathbb{R}^m$, with components $r_j \geq 0$, $\sum_{j=1}^m r_j = 1$, such that the following relations hold:*

$$\dot{x}^*(t) = \frac{\partial}{\partial \lambda} H(x^*(t), u^*(t); \lambda(t); \mathbf{r}) \tag{6}$$

$$\dot{\lambda}(t) = -\frac{\partial}{\partial x} H(x^*(t), u^*(t); \lambda(t); \mathbf{r}) \tag{7}$$

$$x^*(0) = x_o \tag{8}$$

$$\lambda(T) = \sum_{j=1}^m r_j \frac{\partial}{\partial x} G_j(x(T)) \tag{9}$$

with

$$H(x^*(t), u^*(t); \lambda(t); \mathbf{r})$$
$$= \max_{u \in U} H(x^*(t), u; \lambda(t); \mathbf{r})$$

where the weighted Hamiltonian is defined by

$$H(x, u; \lambda; \mathbf{r}) = \sum_{j=1}^m r_j \, g_j(x, u) + \lambda^T \, f(x, u).$$

The proof of this result necessitates some additional regularity assumptions. Some of these conditions imply that there exist differentiable Bellman value functions (see, e.g., Blaquière et al. 1972); some others use the formalism of nonsmooth analysis (see, e.g., Bellassali and Jourani 2004).

The Nash Bargaining Solution

Since Pareto optimal solutions are numerous (actually since a subset of Pareto outcomes are indexed over the weightings $\mathbf{r}$, $r_j > 0$, $\sum_{j=1}^m r_j = 1$), one can expect, in the payoff m-dimensional space, to have a manifold of Pareto outcomes. Therefore, the problem that we must solve now is *how to select the "best" Pareto outcome*? "Best" is a misnomer here, because, by their very definition, two Pareto outcomes cannot be compared or gauged. The choice of a Pareto outcome that satisfies each player must be the result of some bargaining. J. Nash addressed this problem very early, in 1951, using a two-player game setting. He developed an axiomatic approach where he proposed four behavior axioms which, if accepted, would determine a unique choice for the bargaining solution. These axioms are called respectively, (i) invariance to affine transformations of utility representations, (ii) Pareto optimality, (iii) independence of irrelevant alternatives, and (iv) symmetry. Then the bargaining point is the Pareto optimal solution that maximizes the product

$$x^* = \text{argmax}_x (\psi_1(x) - \psi_1(x^\circ))(\psi_2(x) - \psi_2(x^\circ))$$

where x° is the status quo decision, in case bargaining fails, and $(\psi_j(x^\circ))$, $j = 1, 2$ are the payoffs associated with this no-accord decision (this defines the so-called threat point). It has been proved (Binmore et al. 1986) that this

solution could be obtained also as the solution of an auxiliary dynamic game in which a sequence of claims and counterclaims is made by the two players when they bargain.

When extended directly to the context of differential or multistage games, the Nash bargaining solution concept proved to lack the important property of time consistency. This was first noticed in Haurie (1976). Let a dynamic game be defined by Eqs. (1)–(5), with $j = 1, 2$. Suppose the status quo decision, if no agreement is reached at initial state $(t = 0, x(0) = x^o)$, consists in playing an open-loop Nash equilibrium, defined by the controls $u_j^N(\cdot) : [0, T] \to U_j$, $j = 1, 2$ and generating the trajectory $x^N(\cdot) : [0, T] \to \mathbb{R}^n$, with $x^N(0) = x_o$. Now applying the Nash bargaining solution scheme to the data of this differential game played at time $t = 0$ and state $x(0) = x_o$, one identifies a particular Pareto optimal solution, associated with the controls $u^*(\cdot) : [0, T] \to U_j$, $j = 1, 2$ and generating the trajectory $x^*(\cdot) : [0, T] \to \mathbb{R}^n$, with $x^*(0) = x_o$. Now assume the two players renegotiate the agreement to play $u_j^*(\cdot)$ at an intermediate point of the Pareto optimal trajectory $(\tau, x^*(\tau))$, $\tau \in (0, T)$. When computed from that point, the status quo strategies are in general not the same as they were at $(0, x_o)$; furthermore, the shape of the Pareto frontier, when the game is played from $(\tau, x^*(\tau))$, is different from what it is when the game is played at $(0, x_o)$. For these two reasons the bargaining solution at $(\tau, x^*(\tau))$ will not coincide in general with the restriction to the interval $[\tau, T]$ of the bargaining solution from $(0, x_o)$. This implies that the solution concept is not *time consistent*. Using feedback strategies, instead of open-loop ones, does not help, as the same phenomena (change of status quo and change of Pareto frontier) occur in a feedback strategy context.

This shows that the cooperative game solutions proposed in the classical theory of games cannot be applied without precaution in a dynamic setting when players have the possibility to renegotiate agreements at any intermediary point $(t, x^*(t))$ of the bargained solution trajectory.

Cores and C-Optimality in Dynamic Games

Characteristic functions and the associated solution concept of core are important elements in the classical theory of cooperative games. In two papers (Haurie 1975; Haurie and Delfour 1974) the basic definitions and properties of the concept of core in dynamic cooperative games were presented. Consider the multistage system, controlled by a set M of m players and defined by

$$x(k + 1) = f^k(x(k), u_M(k)),$$
$$k = 0, 1, \ldots, K - 1$$
$$x(i) = x^i, \ i \in \{0, 1, \ldots, K - 1\}$$
$$u_M(k) \triangleq (u_j(k))_{j \in M} \in U_M(k) \triangleq \prod_{j \in M} U_j(k).$$

From the initial point (i, x^i) a control sequence $(u_M(i), \ldots, u_M(K - 1))$ generates for each player j a payoff defined as follows:

$$J_j(i, x^i; u_M(i), \ldots, u_M(K - 1)) \triangleq$$
$$\sum_{k=i}^{K-1} \Phi_j(x(k), u_M(k)) + \Upsilon_j(x(K)).$$

A subset S of M is called a coalition. Let $\mu_S^k : x(k) \mapsto u_S(k) \in \prod_{j \in S} U_j(k)$ be a feedback control for the coalition defined at each stage k. A player $j \in S$ considers then, from any initial point (i, x^i), his guaranteed payoff:

$$\Psi_j(i, x^i; \mu_S^i, \ldots, \mu_S^{K-1}) \triangleq$$
$$\inf_{u_{M-s}(i) \in U_{M-s}(i), \ldots, u_{M-s}(K-1) \in U_{M-s}(K-1)}$$
$$\sum_{k=i}^{K-1} \Phi_j(x(k), [\mu_S^k(x(k)), u_{M-S}(k)])$$
$$+ \Upsilon_j(x(K)).$$

Definition 2 The characteristic function at stage i for coalition $S \subset M$ is the mapping $v^i : (S, x^i) \mapsto v^i(S, x^i) \subset \mathbb{R}^S$ defined by

$$\omega_S \triangleq (\omega_j)_{j \in S} \in v^i(S, x^i) \Leftrightarrow$$

$$\exists \mu_S^i, \ldots, \mu_S^{K-1} : \forall j \in S$$

$$\Psi_j(i, x^i; \mu_S^i, \ldots, \mu_S^{K-1}) \geq \omega_j.$$

In other words, there is a feedback law for the coalition S which guarantees at least ω_j to each player j in the coalition.

Suppose that in a cooperative agreement, at point (i, x^i), the coalition S is proposed a gain vector ω_s which is interior to $v^i(S, x^i)$. Then coalition S will block this agreement, because using an appropriate feedback, the coalition can guarantee a better payoff to each of its members. We can now extend the definition of the core of a cooperative game to the context of dynamic games, as the set of agreement gains that cannot be blocked by any coalition.

Definition 3 The core $\Omega(i, x^i)$ at point (i, x^i) is the set of gain vectors $\omega_M \triangleq (\omega_j)_{j \in M}$ such that:
1. There exists a Pareto optimal control $u_M^\star(i), \ldots, u_M^\star(K-1)$ for which $\omega_j = J_j(i, x^i; u_M^\star(i), \ldots, u_M^\star(K-1))$,
2. $\forall S \subset M$ the projection of ω_M in $\mathbb{R}^S$ is not interior to $v^i(S, x^i)$

Playing a cooperative game, one would be interested in finding a solution where the gain-to-go remains in the core at each point of the trajectory. This leads us to define the following.

Definition 4 A control $\tilde{u}^o \triangleq (u_M^o(0), \ldots, u_M (K-1))$ is C-optimal at $(0, x^0)$ if $\tilde{u}^o$ is Pareto optimal generating a state trajectory

$$\{x^o(0) = x^0, x^o(1), \ldots, x^o(K)\}$$

and a sequence of gain-to-go values

$$\omega_j^o(i) = J_j(i, x^o(i); u_M^o(i), \ldots, u_M^o(K-1)),$$
$$i = 0, \ldots, K-1$$

such that $\forall i = 0, 1, \ldots, K-1$, the m-vector $\omega_M^o(i)$ is element of the core $\Omega(i, x^o(i))$.

A C-optimal control generates an agreement which cannot be blocked by any coalition along the Pareto optimal trajectory. It can be shown on

examples that a Pareto optimal trajectory which has the gain-to-go vector in the core at initial point $(0, x_0)$ is not C-optimal.

Links with Reachability Theory for Perturbed Systems

The computation of characteristic functions can be made using the techniques developed to study reachability of dynamic systems with set constrained disturbances (see Bertsekas and Rhodes 1971). Consider the particular case of a linear system

$$x(k+1) = A^k x(k) + \sum_{j \in M} B_j^k u_j(k) \quad (10)$$

where $x \in \mathbb{R}^n$, $u_j \in U_j^k \subset \mathbb{R}^{p_j}$, where U_j^k is a convex-bound set and A^k, B_j^k are matrices of appropriate dimensions. Let the payoff to player j be defined by:

$$J_j(i, x^i; u_M(i), \ldots, u_M(K-1)) \triangleq$$

$$\sum_{k=i}^{K-1} \phi_j^k(x(k)) + \gamma_j^k(u_j(k)) + \Upsilon_j(x(K)).$$

Algorithm Here we use the notations $\phi_S^k \triangleq (\phi_j^k)_{j \in S}$ and $B_S^k u_S \triangleq \sum_{j \in S} B_j^k u_j$. Also we denote $\{u + V\}$, where u is a vector in $\mathbb{R}^m$ and $V \subset \mathbb{R}^m$, the set of vectors $u + v$, $\forall v \in V$. Then
1. $\forall x^K \ v^K(S, x^K) \triangleq \{\omega_S \in \mathbb{R}^S : \Upsilon_S(x^K) \geq \omega_S\}$
2. $\forall x \ \mathcal{E}^{k+1}(S, x) \triangleq \cap v \in U_{M-S} \ v^{k+1}$
   $(S, x + B_{M_S}^k v)$
3. $\forall x^k \ \mathcal{H}^k(S, x^k) \triangleq \underset{u \in U_S}{\cup} \{\gamma_S^k(u) + \mathcal{E}^{k+1}$
   $(S, A^k x^k + B_S^k u)\}$
4. $\forall x^k \ v^k(S, x^k) = \{\phi_S^k(x^k) + \mathcal{H}^k(S, x^k)\}.$
   In an open-loop control setting, the calculation of characteristic function can be done using the concept of Pareto optimal solution for a system with set constrained disturbances, as shown in Goffin and Haurie (1973, 1976) and Haurie (1973).

Conclusion

Since the foundations of a theory of cooperative solutions to dynamic games, recalled in this article, the research has evolved toward the search for cooperative solutions that could be also equilibrium solution, using for that purpose a class of memory strategies Haurie and Towinski (1985), and has found a very important domain of application in the assessment of environmental agreements, in particular those related to the climate change issue. For example, the sustainability of solutions in the core of a dynamic game modeling international environmental negotiations is studied in Germain et al. (2003). A more encompassing model of dynamic formation of coalitions and stabilization of solutions through the use of threats is proposed in Breton et al. (2010). These references are indicative of the trend of research in this field.

Cross-References

▶ Dynamic Noncooperative Games
▶ Game Theory: Historical Overview
▶ Strategic Form Games and Nash Equilibrium

Bibliography

Basile G, Vincent TL (1970) Absolutely cooperative solution for a linear, multiplayer differential game. J Optim Theory Appl 6:41–46

Bellassali S, Jourani A (2004) Necessary optimality conditions in multiobjective dynamic optimization. SIAM J Control Optim 42:2043–2061

Bertsekas DP, Rhodes IB (1971) On the minimax reachability of target sets and target tubes. Automatica 7: 23–247

Binmore K, Rubinstein A, Wolinsky A (1986) The Nash bargaining solution in economic modelling. Rand J Econ 17(2):176–188

Blaquière A, Juricek L, Wiese KE (1972) Geometry of Pareto equilibria and maximum principle in n-person differential games. J Optim Theory Appl 38:223–243

Breton M, Sbragia L, Zaccour G (2010) A dynamic model for international environmental agreements. Environ Resour Econ 45:25–48

Case JH (1969) Toward a theory of many player differential games. SIAM J Control 7(2):179–197

Da Cunha NO, Polak E (1967a) Constrained minimization under vector-valued criteria in linear topological spaces. In: Balakrishnan AV, Neustadt LW (eds) Mathematical theory of control. Academic, New York, pp 96–108

Da Cunha NO, Polak E (1967b) Constrained minimization under vector-valued criteria in finite dimensional spaces. J Math Anal Appl 19:103–124

Germain M, Toint P, Tulkens H, Zeeuw A (2003) Transfers to sustain dynamic core-theoretic cooperation in international stock pollutant control. J Econ Dyn Control 28:79–99

Goffin JL, Haurie A (1973) Necessary conditions and sufficient conditions for Pareto optimality in a multicriterion perturbed system. In: Conti R, Ruberti A (eds) 5th conference on optimization techniques, Rome. Lecture notes in computer science, vol 4 Springer

Goffin JL, Haurie A (1976) Pareto optimality with non-differentiable cost functions. In: Thiriez H, Zionts S (eds) Multiple criteria decision making. Lecture notes in economics and mathematical systems, vol 130. Springer, Berlin/New York, pp 232–246

Haurie A (1973) On Pareto optimal decisions for a coalition of a subset of players. IEEE Trans Autom Control 18:144–149

Haurie A (1975) On some properties of the characteristic function and the core of a multistage game of coalition. IEEE Trans Autom Control 20(2):238–241

Haurie A (1976) A note on nonzero-sum differential games with bargaining solutions. J Optim Theory Appl 13:31–39

Haurie A, Delfour MC (1974) Individual and collective rationality in a dynamic Pareto equilibrium. J Optim Appl 13(3):290–302

Haurie A, Towinski B (1985) Definition and properties of cooperative equilibria in a two-player game of infinite duration. J Optim Theory Appl 46(4):525–534

Isaacs R (1954) Differential games I: introduction. Rand Research Memorandum, RM-1391-30. Rand Corporation, Santa Monica

Leitmann G, Rocklin S, Vincent TL (1972) A note on control space properties of cooperative games. J Optim Theory Appl 9:379–390

Nash J (1950) The bargaining problem. Econometrica 18(2):155–162

Pareto V (1896) Cours d'Economie Politique. Rogue, Lausanne

Salukvadze ME (1971) On the optimization of control systems with vector criteria. In: Proceedings of the 11th all-union conference on control, Part 2. Nauka

Shapley LS (1953) Stochastic games. PNAS 39(10):1095–1100

Starr AW, Ho YC (1969) Nonzero-sum differential games. J Optim Theory Appl 3(3):184–206

Vincent TL, Leitmann G (1970) Control space properties of cooperative games. J Optim Theory Appl 6(2): 91–113

von Neumann J, Morgenstern O (1944) Theory of Games and Economic Behavior, Princeton University Press

Zadeh LA (1963) Optimality and non-scalar-valued performance criteria. IEEE Trans Autom Control AC-8:59–60

Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms

Alejandro D. Domínguez-García[1] and
Christoforos N. Hadjicostis[2]
[1]University of Illinois at Urbana-Champaign,
Urbana-Champaign, IL, USA
[2]University of Cyprus, Nicosia, Cyprus

Abstract

We discuss the utilization of distributed energy resources (DERs) to provide active and reactive power support for ancillary services. Though the amount of active and/or reactive power provided individually by each of these resources can be very small, their presence in large numbers in power distribution networks implies that, under proper coordination mechanisms, they can collectively provide substantial active and reactive power regulation capacity. In this entry, we provide a simple formulation of the DER coordination problem for enabling their utilization to provide ancillary services. We also provide specific architectures and algorithmic solutions to solve the DER coordination problem, with focus on decentralized solutions.

Keywords

Ancillary services; Consensus; Distributed algorithms; Distributed energy resources (DERs)

Introduction

On the distribution side of a power system, there are many distributed energy resources (DERs), e.g., photovoltaic (PV) installations, plug-in hybrid electric vehicles (PHEVs), and thermostatically controlled loads (TCLs), that can be potentially used to provide ancillary services, e.g., reactive power support for voltage control (see, e.g., Turitsyn et al. (2011) and the references therein) and active power up and down regulation for frequency control (see, e.g., Callaway and Hiskens (2011) and the references therein). To enable DERs to provide ancillary services, it is necessary to develop appropriate control and coordination mechanisms. One potential solution relies on a centralized control architecture in which each DER is directly coordinated by (and communicates with) a central decision maker. An alternative approach is to distribute the decision making, which obviates the need for a central decision maker to coordinate the DERs. In both cases, the decision making involves solving a *resource allocation* problem for coordinating the DERs to collectively provide a certain amount of a resource (e.g., active or reactive power).

In a practical setting, whether a centralized or a distributed architecture is adopted, the control of DERs for ancillary services provision will involve some aggregating entity that will gather together and coordinate a set of DERs, which will provide certain amount of active or reactive power in exchange for monetary benefits. In general, these aggregating entities are the ones that interact with the ancillary services market, and through some market-clearing mechanism, they enter a contract to provide some amount of resource, e.g., active and/or reactive power over a period of time. The goal of the aggregating entity is to provide this amount of resource by properly coordinating and controlling the DERs, while ensuring that the total monetary compensation to the DERs for providing the resource is below the monetary benefit that the aggregating entity obtains by selling the resource in the ancillary services market.

In the context above, a household with a solar PV rooftop installation and a PHEV might choose to offer the PV installation to a renewable aggregator so it is utilized to provide reactive power support (this can be achieved as long as the PV installation power electronics-based grid interface has the correct topology Domínguez-García et al. 2011). Additionally, the household could offer its PHEV to a battery vehicle aggregator to be used as a controllable load for energy peak shaving during peak hours and load leveling at night (Guille and Gross 2009).

Finally, the household might choose to enroll in a demand response program in which it allows a demand response provider to control its TCLs to provide frequency regulation services (Callaway and Hiskens 2011). In general, the renewable aggregator, the battery vehicle aggregator, and the demand response provider can be either separate entities or they can be the same entity. In this entry, we will refer to these aggregating entities as *aggregators*.

The Problem of DER Coordination

Without loss of generality, denote by x_j the amount of resource provided by DER i without specifying whether it is active or reactive power. [However, it is understood that each DER provides (or consumes) the same type of resource, i.e., all the x_i's are either active or reactive power.] Let $0 < \underline{x}_i < \overline{x}_i$, for $i = 1, 2, \ldots, n$, denote the minimum ($\underline{x}_i$) and maximum ($\overline{x}_i$) capacity limits on the amount of resource x_i that node i can provide. Denote by X the total amount of resource that the DERs must collectively provide to satisfy the aggregator request. Let $\pi_i(x_i)$ denote the price that the aggregator pays DER i per unit of resource x_i that it provides. Then, the objective of the aggregator in the DER coordination problem is to minimize the total monetary amount to be paid to the DERs for providing the total amount of resource X while satisfying the individual capacity constraints of the DERs. Thus, the DER coordination problem can be formulated as follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} x_i \pi_i(x_i) \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = X \\
& 0 < \underline{x}_i \leq x_i \leq \overline{x}_i, \ \forall j.
\end{aligned} \tag{1}
$$

By allowing heterogeneity in the price per unit of resource that the aggregator offers to each DER, we can take into account the fact that the aggregator might value classes of DERs differently. For example, the downregulation capacity provided by a residential PV installation (which is achieved by curtailing its power) might be valued differently from the downregulation capacity provided by a TCL or a PHEV (both would need to absorb additional power in order to provide downregulation).

It is not difficult to see that if the price functions $\pi_i(\cdot)$, $i = 1, 2 \ldots, n$, are convex and nondecreasing, then the cost function $\sum_{i=1}^{n} x_i \pi_i(x_i)$ is convex; thus, if the problem in (1) is feasible, then there exists a globally optimal solution. Additionally, if the price per unit of resource is linear with the amount of resource, i.e., $\pi_i(x_i) = c_i x_i$, $i = 1, 2, \ldots, n$, then $x_i \pi_i(x_i) = c_i x_i^2$, $i = 1, 2, \ldots, n$, and the problem in (1) reduces to a quadratic program. Also, if the price per unit of resource is constant, i.e., $\pi_i(x_i) = c_i$, $i = 1, 2, \ldots, n$, then $x_i \pi_i(x_i) = c_i x_i$, $i = 1, 2, \ldots, n$, and the problem in (1) reduces to a linear program. Finally, if $\pi_i(x_i) = \pi(x_i) = c$, $i = 1, 2, \ldots, n$, for some constant $c > 0$, i.e., the price offered by the aggregator is constant and the same for all DERs, then the optimization problem in (1) becomes a feasibility problem of the form

$$
\begin{aligned}
\text{find} \quad & x_1, x_2, \ldots, x_n \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = X \\
& 0 < \underline{x}_i \leq x_i \leq \overline{x}_i, \ \forall j.
\end{aligned} \tag{2}
$$

If the problem in (2) is indeed feasible (i.e., $\sum_{l=1}^{n} \underline{x}_l \leq X \leq \sum_{l=1}^{n} \overline{x}_l$), then there is an infinite number of solutions. One such solution, which we refer to as *fair splitting*, is given by

$$
x_i = \underline{x}_i + \frac{X - \sum_{l=1}^{n} \underline{x}_l}{\sum_{l=1}^{n} (\overline{x}_l - \underline{x}_l)} (\overline{x}_i - \underline{x}_i), \ \forall i. \tag{3}
$$

The formulation to the DER coordination problem provided in (2) is not the only possible one. In this regard, and in the context of PHEVs, several recent works have proposed game-theoretic formulations to the problem (Gharesifard et al. 2013; Ma et al. 2013; Tushar et al. 2012). For example, in Gharesifard et al. (2013),
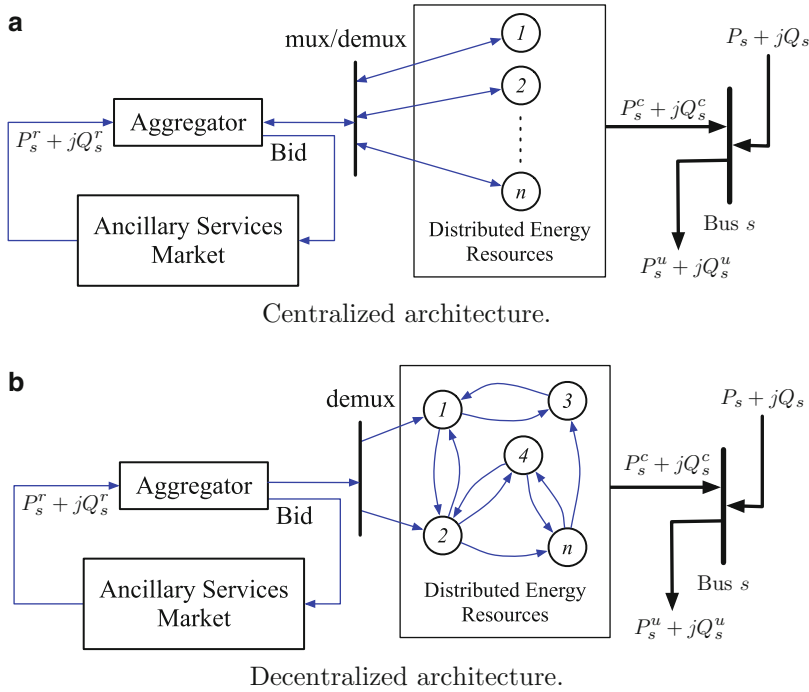
Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms, Fig. 1 Control architecture alternatives. (**a**) Centralized architecture. (**b**) Decentralized architecture

the authors assume that each PHEV is a decision maker and can freely choose to participate after receiving a request from the aggregator. The decision that each PHEV is faced with depends on its own utility function, along with some pricing strategy designed by the aggregator. The PHEVs are assumed to be price anticipating in the sense that they are aware of the fact that the pricing is designed by the aggregator with respect to the average energy available. Another alternative is to formulate the DER coordination problem as a scheduling problem (Chen et al. 2012; Subramanian et al. 2012), where the DERs are treated as tasks. Then, the problem is to develop real-time scheduling policies to service these tasks.

Architectures

Next, we describe two possible architectures that can be utilized to implement the proper algorithms for solving the DER coordination problem

as formulated in (1). Specifically, we describe a centralized architecture that requires the aggregator to communicate bidirectionally with each DER and a distributed architecture that requires the aggregator to only unidirectionally communicate with a limited number of DERs but requires some additional exchange of information (not necessarily through bidirectional communication links) among the DERs.

Centralized Architecture

A solution can be achieved through the completely centralized architecture of Fig. 1a, where the aggregator can exchange information with each available DER. In this scenario, each DER can inform the aggregator about its active and/or reactive capacity limits and other operational constraints, e.g., maintenance schedule. After gathering all this information, the aggregator solves the optimization program in (1), the solution of which will determine how to allocate among the resources the total amount of active power P_s^r and/or reactive

power Q_s^r that it needs to provide. Then, the aggregator sends individual commands to each DER so they modify their active and or reactive power generation according to the solution of (1) computed by the aggregator. In this centralized solution, however, it is necessary to overlay a communication network connecting the aggregator with each resource and to maintain knowledge of the resources that are available at any given time.

Decentralized Architecture

An alternative is to use the decentralized control architecture of Fig. 1b, where the aggregator relays information to a limited number of DERs that it can directly communicate with and each DER is able to exchange information with a number of other close-by DERs. For example, the aggregator might broadcast the prices to be paid to each type of DER. Then, through some distributed protocol that adheres to the communication network interconnecting the DERs, the information relayed by the aggregator to this limited number of DERs is disseminated to all other available DERs. This dissemination process may rely on flooding algorithms, message-passing protocols, or linear-iterative algorithms as proposed in Domínguez-García and Hadjicostis (2010, 2011). After the dissemination process is complete and through a distributed computation over the communication network, the DERs can solve the optimization program in (1) and determine its active and/or reactive power contribution.

A decentralized architecture like the one in Fig. 1b may offer several advantages over the centralized one in Fig. 1b, including the following. First, a decentralized architecture may be more economical because it does not require communication between the aggregator and the various DERs. Also, a decentralized architecture does not require the aggregator to have a complete knowledge of the DERs available. Additionally, a decentralized architecture can be more resilient to faults and/or unpredictable behavioral patterns by the DERs. Finally, the practical implementation of such decentralized architecture can rely on inexpensive and simple hardware. For example,

the testbed described in Domínguez-García et al. (2012a), which is used to solve a particular instance of the problem in (1), uses Arduino microcontrollers (see Arduino for a description) outfitted with wireless transceivers implementing a ZigBee protocol (see ZigBee for a description).

Algorithms

Ultimately, whether a centralized or a decentralized architecture is adopted, it is necessary to solve the optimization problem in (1). If a centralized architecture is adopted, then solving (1) is relatively straightforward using, e.g., standard gradient-descent algorithms (see, e.g., Bertsekas and Tsitsiklis 1997). Beyond the DER coordination problem and the specific formulation in (1), solving an optimization problem is challenging if a decentralized architecture is adopted (especially if the communication links between DERs are not bidirectional); this has spurred significant research in the last few years (see, e.g., Bertsekas and Tsitsiklis 1997, Xiao et al. 2006, Nedic et al. 2010, Zanella et al. 2011, Gharesifard and Cortes 2012, and the references therein).

In the specific context of the DER coordination problem as formulated in (1), when the cost functions are assumed to be quadratic and the communication between DERs is not bidirectional, an algorithm amenable for implementation in a decentralized architecture like the one in Fig. 1b has been proposed in Domínguez-García et al. (2012a). Also, in the context of Fig. 1b, when the communication between DERs are bidirectional, the DER coordination problem, as formulated in (1), can be solved using an algorithm proposed in Kar and Hug (2012).

As mentioned earlier, when the price offered by the aggregator is constant and identical for all DERs, the problem in (1) reduces to the feasibility problem in (2). One possible solution to this feasibility problem is the fair-splitting solution in (3). Next, we describe a linear-iterative algorithm – originally proposed in Domínguez-García and Hadjicostis (2010, 2011) and referred to as *ratio consensus* – that allows the DERs to

individually determine its contribution so that the fair-splitting solution is achieved.

Ratio Consensus: A Distributed Algorithm for Fair Splitting

We assume that each DER is equipped with a processor that can perform simple computations and can exchange information with neighboring DERs. In particular, the information exchange between DERs can be described by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the vertex set (each vertex – or node – corresponds to a DER) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, where $(i, j) \in \mathcal{E}$ if node i can receive information from node j. We require $\mathcal{G}$ to be *strongly connected*, i.e., for any pair of vertices l and l', there exists a path that starts in l and ends in l'. Let $\mathcal{L}^+ \subseteq \mathcal{V}, \mathcal{L}^+ \neq \emptyset$ denote the set of nodes that the aggregator is able to directly communicate with.

The processor of each DER i maintains two values y_i and z_i, which we refer to as internal states, and updates them (independently of each other) to be, respectively, a linear combination of DER i's own previous internal states and the previous internal states of all nodes that can possibly transmit information to node i (including itself). In particular, for all $k \geq 0$, each node i updates its two internal states as follows:

$$y_i[k+1] = \sum_{j \in \mathcal{N}_i^-} \frac{1}{\mathcal{D}_j^+} y_j[k], \qquad (4)$$

$$z_i[k+1] = \sum_{j \in \mathcal{N}_i^-} \frac{1}{\mathcal{D}_j^+} z_j[k], \qquad (5)$$

where $\mathcal{N}_i^- = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, i.e., all nodes that can possibly transmit information to node i (including itself); and $\mathcal{D}_i^+$ is the out-degree of node i, i.e., the number of nodes to which node i can possibly transmit information (including itself). The initial conditions in (4) are set to $y_i[0] = X/m - \underline{x}_i$ if $i \in \mathcal{L}^+$, and $y_i[0] = -\underline{x}_i$ otherwise and the initial conditions in (5) are set to $z_i[0] = \overline{x}_i - \underline{x}_i$. Then, as shown in Domínguez-García and Hadjicostis (2011), as long as $\sum_{l=1}^n \overline{x}_l \leq X \leq \sum_{l=1}^n \overline{x}_l$, each DER i can asymptotically calculate its contribution as

$$x_i = \underline{x}_i + \gamma(\overline{x}_i - \underline{x}_i) \qquad (6)$$

where for all i

$$\lim_{k \to \infty} \frac{y_i[k]}{z_i[k]} = \frac{X - \sum_{l=1}^n \underline{x}_l}{\sum_{l=1}^n (\overline{x}_l - \underline{x}_l)} := \gamma. \qquad (7)$$

It is important to note that the algorithm in (4)–(7) also serves as a primitive for the algorithm proposed in Domínguez-García et al. (2012a), which solves the problem in (1) when the cost function is quadratic. Also, the algorithm in (4)–(7) is not resilient to packet-dropping communication links or imperfect synchronization among the DERs, which makes it difficult to implement in practice; however, there are robustified variants of this algorithm that address these issues Domínguez-García et al. (2012b) and have been demonstrated to work in practice (Domínguez-García et al. 2012a).

Cross-References

Bibliography

Arduino [Online]. Available: http://www.arduino.cc

Bertsekas DP, Tsitsiklis JN (1997) Parallel and distributed computation. Athena Scientific, Belmont

Callaway DS, Hiskens IA (2011) Achieving controllability of electric loads. Proc IEEE 99(1):184–199

Chen S, Ji Y, Tong L (2012) Large scale charging of electric vehicles. In: Proceedings of the IEEE power and energy society general meeting, San Diego

Domínguez-García AD, Hadjicostis CN (2010) Coordination and control of distributed energy resources for provision of ancillary services. In: Proceedings of the IEEE SmartGridComm, Gaithersburg

Domínguez-García AD, Hadjicostis CN (2011) Distributed algorithms for control of demand response and distributed energy resources. In: Proceedings of the IEEE conference on decision and control, Orlando

Domínguez-García AD, Hadjicostis CN, Krein PT, Cady ST (2011) Small inverter-interfaced distributed energy resources for reactive power support. In: Proceedings of the IEEE applied power electronics conference and exposition, Fort Worth

Domínguez-García AD, Cady ST, Hadjicostis CN (2012a) Decentralized optimal dispatch of distributed energy resources. In: Proceedings of the IEEE conference on decision and control, Maui

Domínguez-García AD, Hadjicostis CN, Vaidya N (2012b) Resilient networked control of distributed energy resources. IEEE J Sel Areas Commun 30(6):1137–1148

Gharesifard B, Cortes J (2012) Continuous-time distributed convex optimization on weight-balanced digraphs. In: Proceedings of the IEEE conference on decision and control, Maui

Gharesifard B, Domínguez-García AD, Başar T (2013) Price-based distributed control for networked plug-in electric vehicles. In: Proceedings of the American control conference, Washington, DC

Guille C, Gross G (2009) A conceptual framework for the vehicle-to-grid (V2G) implementation. Energy Policy 37(11):4379–4390

Kar S, Hug G (2012) Distributed robust economic dispatch in power systems: a consensus + innovations approach. In: Proceedings of the IEEE power and energy society general meeting, San Diego

Ma Z, Callaway DS, Hiskens IA (2013) Decentralized charging control of large populations of plug-in electric vehicles. IEEE Trans Control Syst Technol 21:67–78

Nedic A, Ozdaglar A, Parrilo PA (2010) Constrained consensus and optimization in multi-agent networks. IEEE Trans Autom Control 55(4):922–938

Subramanian A, Garcia M, Domínguez-García AD, Callaway DC, Poolla K, Varaiya P (2012) Real-time scheduling of deferrable electric loads. In: Proceedings of the American control conference, Montreal

Turitsyn K, Sulc P, Backhaus S, Chertkov M (2011) Options for control of reactive power by distributed photovoltaic generators. Proc IEEE 99(6):1063–1073

Tushar W, Saad W, Poor HV, Smith DB (2012) Economics of electric vehicle charging: a game theoretic approach. IEEE Trans Smart Grids 3(4):1767–1778

Xiao L, Boyd S, Tseng CP (2006) Optimal scaling of a gradient method for distributed resource allocation. J Optim Theory Appl 129(3):469–488

Zanella F, Varagnolo D, Cenedese A, Pillonetto G, Schenato L (2011) Newton-Raphson consensus for distributed convex optimization. In: Proceedings of the IEEE conference on decision and control, Orlando

ZigBee Alliance [Online]. Available: http://www.zigbee.org

Credit Risk Modeling

Tomasz R. Bielecki
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL, USA

Abstract

Modeling of credit risk is concerned with constructing and studying formal models of time evolution of credit ratings (credit migrations) in a pool of credit names, and with studying various properties of such models. In particular, this involves modeling and studying default times and their functionals.

Keywords

Credit risk; Credit migrations; Default time; Markov copulae

Introduction

Modeling of credit risk is concerned with constructing and studying formal models of time evolution of credit ratings (credit migrations) in a pool of N credit names (obligors), and with studying various properties of such models. In particular, this involves modeling and studying default times and their functionals. In many ways, modeling techniques used in credit risk are similar to modeling techniques used in reliability theory. Here, we focus on modeling in continuous time.

Models of credit risk are used for the purpose of valuation and hedging of credit derivatives, for valuation and hedging of counter-party risk, for assessment of systemic risk in an economy, or for constructing optimal trading strategies involving credit-sensitive financial instruments, among other uses.

Evolution of credit ratings for a single obligor, labeled as i, where $i \in \{1, \ldots, N\}$, can be

modeled in many possible ways. One popular possibility is to model credit migrations in terms of a jump process, say $C^i = (C^i_t)_{t \geq 0}$, taking values in a finite set, say $\mathcal{K}^i := \{0, 1, 2, \ldots, K^i - 1, K^i\}$, representing credit ratings assigned to obligor i. Typically, the rating state K^i represents the state of default of the i-th obligor, and typically it is assumed that process C^i is absorbed at state K^i.

Frequently, the case when $K^i = 1$, that is $\mathcal{K}^i := \{0, 1\}$, is considered. In this case, one is only concerned with jump from the pre-default state 0 to the default state 1, which is usually assumed to be absorbing – the assumption made here as well. It is assumed that process C^i starts from state 0. The (random) time of jump of process C^i from state 0 to state 1 is called the default time, and is denoted as τ^i. Process C^i is now the same as the indicator process of τ^i, which is denoted as H^i and defined as $H^i_t = \mathbb{1}_{\{\tau^i \leq t\}}$, for $t \geq 0$. Consequently, modeling of the process C^i is equivalent to modeling of the default time τ^i.

The ultimate goal of credit risk modeling is to provide a feasible mathematical and computational methodology for modeling the evolution of the multivariate credit migration process $\mathbf{C} := (C^1, \ldots, C^N)$, so that relevant functionals of such processes can be computed efficiently. The simplest example of such functional is $P(\mathbf{C}_{t_j} \in A_j, j = 1, 2, \ldots, J | \mathcal{G}_s)$, representing the conditional probability, given the information $\mathcal{G}_s$ at time $s \geq 0$, that process $\mathbf{C}$ takes values in the set A_j at time $t_j \geq 0$, $j = 1, 2, \ldots, J$. In particular, in case of modeling of the default times τ^i, $i = 1, 2, \ldots, N$, one is concerned with computing conditional survival probabilities $P(\tau^1 > t_1, \ldots, \tau^N > t_N | \mathcal{G}_s)$, which are the same as probabilities $P(H^i_{t_i} = 0, i = 1, 2, \ldots, N | \mathcal{G}_s)$.

Based on that, one can compute more complicated functionals, that naturally occur in the context of valuation and hedging of credit risk–sensitive financial instruments, such as corporate (defaultable) bonds, credit default swaps, credit spread options, collateralized bond obligations, and asset-based securities, for example.

Modeling of Single Default Time Using Conditional Density

Traditionally, there were two main approaches to modeling default times: the structural approach and the reduced approach, also known as the hazard process approach. The main features of both these approaches are presented in Bielecki and Rutkowski (2004).

We focus here on modeling a single default time, denoted as τ, using the so-called conditional density approach of El Karoui et al. (2010). This approach allows for extension of results that can be derived using reduced approach.

The default time τ is a strictly positive random variable defined on the underlying probability space $(\Omega, \mathcal{F}, P)$, which is endowed with a reference filtration, say $\mathbb{F} = (\mathcal{F}_t)_{t \geq 0}$, representing flow of all (relevant) market information available in the model, not including information about occurrence of τ. The information about occurrence of τ is carried by the (right continuous) filtration $\mathbb{H}$ generated by the indicator process $H := (H_t = \mathbb{1}_{\tau \leq t})_{t \geq 0}$. The full information in the model is represented by filtration $\mathbb{G} := \mathbb{F} \vee \mathbb{H}$.

It is postulated that

$$P(\tau \in d\theta | \mathcal{F}_t) = \alpha_t(\theta)d\theta,$$

for some random field $\alpha.(\cdot)$, such that $\alpha_t(\cdot)$ is $\mathcal{F}_t \otimes \mathcal{B}(\mathbb{R}_+)$ measurable for each t. The family $\alpha_t(\cdot)$ is called $\mathcal{F}_t$-conditional density of τ. In particular, $P(\tau > \theta) = \int_\theta^\infty \alpha_0(u) \, du$. The following survival processes are associated with τ,

- $S_t(\theta) := P(\tau > \theta | \mathcal{F}_t) = \int_\theta^\infty \alpha_t(u) \, du$, which is an $\mathbb{F}$-martingale,
- $S_t := S_t(t) = P(\tau > t | \mathcal{F}_t)$, which is an $\mathbb{F}$-supermartingale (Azéma supermartingale).

In particular, $S_0(\theta) = P(\tau > \theta) = \int_\theta^\infty \alpha_0(u) \, du$, and $S_t(0) = S_0 = 1$.

As an example of computations that can be done using the conditional density approach we give the following result, in which notation "bd" and "ad" stand for `before default` and `at-or-after default`, respectively.

Theorem 1 *Let $Y_T(\tau)$ be a $\mathcal{F}_T \vee \sigma(\tau)$ measurable and bounded random variable. Then*

$$E(Y_T(\tau)|\mathcal{F}_t) = Y_t^{\mathrm{bd}}\mathbb{1}_{t < \tau} + Y_t^{\mathrm{ad}}(T, \tau)\mathbb{1}_{t \geq \tau},$$

where

$$Y_t^{\mathrm{bd}} = \frac{\int_t^\infty Y_T(\theta)\alpha_t(\theta)d\theta}{S_t}\mathbb{1}_{S_t > 0},$$

and

$$Y_t^{\mathrm{ad}}(T, \theta) = \frac{E(Y_T(\theta)\alpha_T(\theta)|\mathcal{F}_t)}{\alpha_t(\theta)}\mathbb{1}_{\alpha_t(\theta) > 0}.$$

There is an interesting connection between the conditional density process and the so-called default intensity processes, which are ones of the main objects used in the reduced approach. This connection starts with the following result,

Theorem 2 *(i) The Doob-Meyer (additive) decomposition of the survival process S is given as*

$$S_t = 1 + M_t^{\mathbb{F}} - \int_0^t \alpha_u(u)du,$$

where $M_t^{\mathbb{F}} = -\int_0^t (\alpha_t(u) - \alpha_u(u))du = E(\int_0^\infty \alpha_u(u)du|\mathcal{F}_t) - 1$.

(ii) Let $\xi := \inf\{t \geq 0 : S_{t-} = 0\}$. Define $\lambda_t^{\mathbb{F}} = \frac{\alpha_t(t)}{S_t}$ for $t < \xi$ and $\lambda_t^{\mathbb{F}} = \lambda_\xi^{\mathbb{F}}$ for $t \geq \xi$. Then, the multiplicative decomposition of S is given as

$$S_t = L_t^{\mathbb{F}} e^{-\int_0^t \lambda_u^{\mathbb{F}}du},$$

where

$$dL_t^{\mathbb{F}} = e^{\int_0^t \lambda_u^{\mathbb{F}}du}dM_t^{\mathbb{F}}, \quad L_0^{\mathbb{F}} = 1.$$

The process $\lambda^{\mathbb{F}}$ is called the $\mathbb{F}$ intensity of τ.

The $\mathbb{G}$-compensator of τ is the $\mathbb{G}$-predictable increasing process $\Lambda^{\mathbb{G}}$ such that the process

$$M_t^{\mathbb{G}} = H_t - \Lambda_t^{\mathbb{G}}$$

is a $\mathbb{G}$-martingale. If $\Lambda^{\mathbb{G}}$ is absolutely continuous, the $\mathbb{G}$-adapted process $\lambda^{\mathbb{G}}$ such that

$$\Lambda_t^{\mathbb{G}} = \int_0^t \lambda_u^{\mathbb{G}}du$$

is called the $\mathbb{G}$-intensity of τ. The $\mathbb{G}$-compensator is stopped at τ, i.e., $\Lambda_t^{\mathbb{G}} = \Lambda_{t \wedge \tau}^{\mathbb{G}}$. Hence, $\lambda_t^{\mathbb{G}} = 0$ when $t > \tau$. In particular, we have

$$\lambda_t^{\mathbb{G}} = \mathbb{1}_{t < \tau}\lambda_t^{\mathbb{F}} = (1 - H_t)\lambda_t^{\mathbb{F}}.$$

The conditional density process and the $\mathbb{G}$-intensity of τ are related as follows: For any $t < \xi$ and $\theta \geq t$ we have

$$\alpha_t(\theta) = E(\lambda_\theta^{\mathbb{G}}|\mathcal{F}_t).$$

Example 1 This is a structural-model-like example
- Suppose $\mathbb{F} = \mathbb{F}^X$ is a filtration of a default driver process, say X, and Θ is the default barrier assumed to be independent of X. Denote $G(t) = P(\Theta > t)$.
- Define

$$\tau := \inf\{t \geq 0 : \Gamma_t \geq \Theta\},$$

with $\Gamma_t := \sup_{s \leq t} X_s$. We then have $S_t(\theta) = G(\Gamma_\theta)$ if $\theta \leq t$ and $S_t(\theta) = E(G(\Gamma_\theta)|\mathcal{F}_t^X)$ if $\theta > t$
- Assume that $F = 1 - G$ and Γ are absolutely continuous w.r.t. Lebesgue measure, with respective densities f and γ. We then have

$$\alpha_t(\theta) = f(\Gamma_\theta)\gamma_\theta = \alpha_\theta, \, t \geq \theta,$$

and $\mathbb{F}^X$ intensity of τ is

$$\lambda_t = \frac{\alpha_t(t)}{G(\Gamma_t)} = \frac{\alpha_t(t)}{S_t}.$$

- In particular, if Θ is a unit exponential r.v., that is, if $G(t) = e^{-t}$ for $t \geq 0$, then we have that $\lambda_t = \gamma_t = \frac{\alpha_t(t)}{S_t}$.

Example 2 This is a reduced-form-like example.

- Suppose S is a strictly positive process. Then, the $\mathbb{F}$-hazard process of τ is denoted by $\Gamma^{\mathbb{F}}$ and is given as

$$\Gamma_t^{\mathbb{F}} = -\ln S_t, \quad t \geq 0.$$

In other words,

$$S_t = e^{-\Gamma_t^{\mathbb{F}}}, \quad t \geq 0.$$

- In particular, if $\Gamma^{\mathbb{F}}$ is absolutely continuous, that is, $\Gamma_t^{\mathbb{F}} = \int_0^t \gamma_u^{\mathbb{F}} \, du$ then

$$S_t = e^{-\int_0^t \gamma_u^{\mathbb{F}} \, du}, \quad t \geq 0 \quad \text{and}$$

$$\alpha_t(\theta) = \gamma_\theta^{\mathbb{F}} S_\theta, \; t \geq \theta.$$

Modeling Evolution of Credit Ratings Using Markov Copulae

The key goal in modeling of the joint migration process $\mathbf{C}$ is that the distributional laws of the individual migration components C^i, $i \in \{1, \ldots, N\}$, agree with given (predetermined) laws. The reason for this is that the marginal laws of $\mathbf{C}$, that is, the laws of C^i, $i \in \{1, \ldots, N\}$, can be calibrated from market quotes for prices of individual (as opposed to basket) credit derivatives, such as the credit default swaps, and thus, the marginals of $\mathbf{C}$ should have laws agreeing with the market data.

One way of achieving this goal is to model $\mathbf{C}$ as a Markov chain satisfying the so-called Markov copula property. For brevity we present here the simplest such model, in which the reference filtration $\mathbb{F}$ is trivial, assuming additionally, but without loss of generality, that $N = 2$ and that $\mathcal{K}^1 = \mathcal{K}^2 = \mathcal{K} := \{0, 1, \ldots, K\}$.

Here we focus on the case of the so-called strong Markov copula property, which is reflected in Theorem 3.

Let us consider two Markov chains Z^1 and Z^2 on $(\Omega, \mathcal{F}, P)$, taking values in a finite state space $\mathcal{K}$, and with the infinitesimal generators $A^1 := [a_{ij}^1]$ and $A^2 := [a_{hk}^2]$, respectively.

Consider the system of linear algebraic equations in unknowns $a_{ih,jk}^{\mathbf{C}}$,

$$\sum_{k \in \mathcal{K}} a_{ih,jk}^{\mathbf{C}} = a_{ij}^1, \quad \forall i, j, h \in \mathcal{K}, \; i \neq j, \quad (1)$$

$$\sum_{j \in \mathcal{K}} a_{ih,jk}^{\mathbf{C}} = a_{hk}^2, \quad \forall i, h, k \in \mathcal{K}, \; h \neq k, \quad (2)$$

It can be shown that this system admits at least one positive solution.

Theorem 3 *Consider an arbitrary positive solution of the system* (1)–(2)*. Then the matrix* $A^{\mathbf{C}} = [a_{ih,jk}^X]_{i,h,j,k \in \mathcal{K}}$ *(where diagonal elements are defined appropriately) satisfies the conditions for a generator matrix of a bivariate time-homogeneous Markov chain, say* $\mathbf{C} = (C^1, C^2)$*, whose components are Markov chains in the filtration of* $\mathbf{C}$ *and with the same laws as* Z^1 *and* Z^2*.*

Consequently, the system (1)–(2) serves as a Markov copula between the Markovian margins C^1, C^2 and the bivariate Markov chain $\mathbf{C}$.

Note that the system (1)–(2) can contain more unknowns than the number of equations, therefore being underdetermined, which is a crucial feature for ability of calibration of the joint migration process $\mathbf{C}$ to marginal market data.

Example 3 This example illustrates modeling joint defaults using strong Markov copula theory.

Let us consider two processes, Z^1 and Z^2, that are time-homogeneous Markov chains, each taking values in the state space $\{0, 1\}$, with respective generators

$$A^1 = \begin{matrix} \\ 0 \\ 1 \end{matrix} \begin{matrix} 0 & \quad 1 \\ \begin{pmatrix} -(a+c) & a+c \\ 0 & 0 \end{pmatrix} \end{matrix} \quad (3)$$

and

$$A^2 = \begin{matrix} \\ 0 \\ 1 \end{matrix} \begin{matrix} 0 & \quad 1 \\ \begin{pmatrix} -(b+c) & b+c \\ 0 & 0 \end{pmatrix} \end{matrix}, \quad (4)$$

for $a, b, c \geq 0$.

The off-diagonal elements of the matrix $A^{\mathbf{C}}$ below satisfy the system (1)–(2),

$$A^{\mathbf{C}} = \begin{array}{c} \\ (0,0) \\ (0,1) \\ (1,0) \\ (1,1) \end{array} \begin{array}{cccc} (0,0) & (0,1) & (1,0) & (1,1) \\ \begin{pmatrix} -(a+b+c) & b & a & c \\ 0 & -(a+c) & 0 & a+c \\ 0 & 0 & -(b+c) & b+c \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{5}$$

Thus, matrix $A^{\mathbf{C}}$ generates a Markovian joint migration process $\mathbf{C} = (C^1, C^2)$, whose components C^1 and C^2 model individual default with prescribed default intensities $a + c$ and $b + c$, respectively.

For more information about Markov copulae and about their applications in credit risk we, refer to Bielecki et al. (2013).

Summary and Future Directions

The future directions in development and applications of credit risk models are comprehensively laid out in the recent volume Bielecki et al. (2011). One additional future direction is modeling of systemic risk.

Cross-References

▶ Financial Markets Modeling
▶ Option Games: The Interface Between Optimal Stopping and Game Theory

Bibliography

We do not give a long list of recommended reading here. That would be in any case incomplete. Up–to–date references can be found on www.defaultrisk.com.

Bielecki TR, Rutkowski M (2004) Credit risk: modeling, valuation and hedging. Springer, Berlin

Bielecki TR, Brigo D, Patras F (eds) (2011) Credit risk frontiers: subprime crisis, pricing and hedging, CVA, MBS, ratings and liquidity. Wiley, Hoboken

Bielecki TR, Jakubowski J, Niewęgłowski M (2013) Intricacies of dependence between components of multivariate Markov chains: weak Markov consistency and Markov copulae. Electron J Probab 18(45):1–21

Bluhm Ch, Overbeck L, Wagner Ch (2010) An introduction to credit risk modeling. Chapman & Hall, Boca Raton

El Karoui N, Jeanblanc M, Jiao Y (2010) What happens after a default: the conditional density approach. SPA 120(7):1011–1032

Schönbucher PhJ (2003) Credit derivatives pricing models. Wiley Finance, Chichester

D

Data Association

Yaakov Bar-Shalom and Richard W. Osborne
University of Connecticut, Storrs, CT, USA

Abstract

In tracking applications, following the signal detection process that yields measurements, there is a procedure that *selects* the measurement(s) to be incorporated into the state estimator – this is called **data association (DA)**. In multitarget-multisensor tracking systems, there are generally three classes of data association: specifically, **measurement-to-track association (M2TA)**, **track-to-track association (T2TA)**, and **measurement-to-measurement association (M2MA)**. M2TA is the process of associating each measurement from a list (originating from one or more sensors) to a new or existing track. T2TA is the process of associating multiple existing tracks (from multiple sensors or from different periods in time), generally with the intent of fusing them afterward. M2MA is the process of associating measurements from different sensors in order to form "composite measurements" and/or do track initialization. The processes of M2TA and T2TA will be discussed in more detail here, while details on M2MA can be found in Bar-Shalom et al. (2011).

Keywords

Clutter; Measurement origin uncertainty; Measurement validation; Persistent interference; Tracking

Introduction

In a radar the "return" from the target of interest is sought within a time interval determined by the anticipated range of the target when it reflects the energy transmitted by the radar: a "range gate" is set up and the detection(s) within this gate can be *associated* with the target of interest.

In general the measurements have a higher dimension:

- Range, azimuth (bearing), elevation, or direction sines for radar, possibly also range rate
- Bearing and frequency (when the signal is narrow band) or time difference of arrival and frequency difference in passive sonar
- Two line-of-sight angles or direction sines for optical or passive electromagnetic sensors

Then a *multidimensional* **gate** is set up for detecting the signal from the target. This is done to avoid searching for the signal from the target of interest in the entire measurement space. A measurement in the gate, while not guaranteed to have originated from the target the gate pertains to, is a *valid association candidate* – thus, the name **validation region** or **association region**. If there

is more than one detection (measurement) in the gate, this leads to an **association uncertainty**.

In the discussion to follow, it will be assumed that one has **point measurements** rather than distributed over several **resolution cells** of the sensor as in the case of an **extended target**.

Similar validation has to be carried out in T2TA.

Validation Region

In view of the variety of variables that can be measured, a generic gating (or validation or association) procedure for continuous-valued measurements is discussed.

Consider a target that is in track, i.e., its filter has been initialized. Then, according to Sect. 5.2.3 of Bar-Shalom et al. (2001), one has the predicted value (mean) of the measurement $\hat{z}(k+1|k)$ and the associated covariance $S(k+1)$.

Assumption. The true measurement conditioned on the past is **normally (Gaussian) distributed** (The notation $\mathcal{N}(x; \mu, S)$ stands for the normal (Gaussian) pdf with the argument (vector) random variable x, mean μ, and covariance matrix S. The reason for the use of the designation "normal" is to distinguish this omnipresent pdf from all the others (abnormal).) with its **probability density function (pdf)** given by

$$p[z(k+1)|Z^k] = \mathcal{N}[z(k+1); \hat{z}(k+1|k),$$
$$S(k+1)] \quad (1)$$

where $S(k+1)$ is the **innovation (residual) covariance** matrix and z is the true measurement.

Then the true measurement will be in the following region:

$$\mathcal{V}(k+1, \gamma) = \{z : d^2 \leq \gamma\} \quad (2)$$

with probability determined by the **gate threshold** γ and

$$d^2 \triangleq [z - \hat{z}(k+1|k)]' S(k+1)^{-1} [z - \hat{z}(k+1|k)] \quad (3)$$

This distance metric, d^2, is referred to in the literature as the **normalized innovation squared (NIS)**, **statistical distance squared**, **Mahalanobis distance**, or **chi-square distance**.

The region defined by (2) is called the **gate** or **validation region** (hence, the notation $\mathcal{V}$) or **association region**. It is also known as the **ellipse (or ellipsoid) of probability concentration** – the region of *minimum volume* that contains a given probability mass under the Gaussian assumption. The semiaxes of the ellipsoid (2) are the square roots of the eigenvalues of γS. The threshold γ is obtained from tables of the chi-square distribution since the quadratic form (3) that defines the validation region in (2) is chi-square distributed with number of degrees of freedom equal to the dimension n_z of the measurement.

Table 1 gives the **gate probability** (The notation $P\{\cdot\}$ is used to denote the probability of event $\{\cdot\}$.)

$$P_G \triangleq P\{z(k+1) \in \mathcal{V}(k+1, \gamma)\} \quad (4)$$

or the "probability that the (true) measurement will fall in the gate" for various values γ and dimensions n_z of the measurement. The square root $g = \sqrt{\gamma}$ is sometimes referred to as the "number of sigmas" (standard deviations) of the gate. This, however, does not fully define the probability mass in the gate as can be seen from Table 1.

Remark 1 It should be pointed out that thresholding in a detector is also a form of gating – only a signal above a certain intensity level (at the end of the signal processing chain) is accepted as a detection and then one has a measurement. In this case the "gate" is the interval $[\tau, \infty]$ in the signal intensity space, where τ is the detection threshold.

A Single Target in Clutter

The validation procedure limits the region in the measurement space where the information processor will "look" to find the measurement from the target of interest. In spite of this, it can happen that *more than one detection*, i.e., several measurements, will be found in the validation region.

Data Association, Table 1 Gate thresholds and the probability mass P_G in the gate

	γ	1	4	6.6	9	9.2	11.4	16	25
	g	1	2	2.57	3	3.03	3.38	4	5
n_z									
1		0.683	0.954	0.99	0.997			0.99994	1
2		0.393	0.865		0.989	0.99		0.9997	1
3		0.199	0.739		0.971		0.99	0.9989	0.99998

Measurements *outside the validation region* can be ignored: they are "too far" and thus very unlikely to have originated from the target of interest. This holds if the gate probability is close to unity and *the model used to obtain the gate is correct*.

The problem of tracking a single target in clutter considers the situation where there are possibly several measurements in the validation region (gate) of a target. The set of **validated measurements** consists of:

- The correct measurement (if detected and it fell in the gate)
- The undesirable measurements: clutter or false alarm originated

In practice detections are obtained by thresholding the signal received by the sensor after processing it. This is the simplest (binary) way of using a target **feature** – its intensity. More sophisticated ways of using such feature information can be found in Bar-Shalom et al. (2011).

It is assumed that the measurement contains all the information that could be used to discard the undesirable measurements. Therefore, any measurement that has been validated could have originated from the target of interest.

A situation with a single-target track and several validated measurements is depicted in Fig. 1. The (two-dimensional) validation region is an ellipse centered at the predicted measurement $\hat{z}^1$. The parameters of the ellipse are determined by the covariance matrix S of the innovation, which is assumed to be Gaussian.

All the measurements in the validation region can be said to be *not too unlikely* to have originated from the target of interest, even though only one is assumed to be the true one.

The implication of the assumption that there is a single target is that the *undesirable measurements constitute a random interference*.
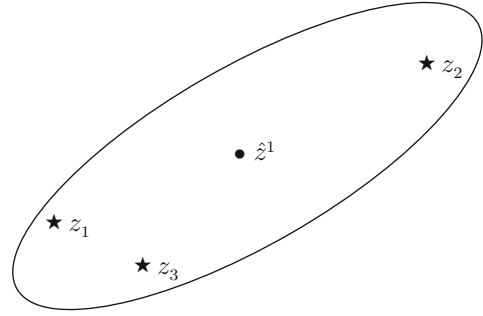
Data Association, Fig. 1 Several measurements in the validation region of a single track

The common mathematical model for such **false measurements** is that they are:

- Uniformly spatially distributed
- Independent across time

This corresponds to what is known as **residual clutter** – the constant clutter, if any, is assumed to have been removed.

Multiple Targets in Clutter

The situation where there are several target tracks in the same neighborhood as well as clutter (or false alarms) is more complicated. Figure 2 illustrates such a case for a given time, with the predicted measurements for the two targets considered denoted as $\hat{z}^1$ and $\hat{z}^2$. In this figure the following measurement origins are possible:

- z_1 from target 1 or clutter
- z_2 from either target 1 or target 2 or clutter
- z_3 and z_4 from target 2 or clutter

However, if z_2 originated from target 2, then it is quite likely that z_1 originated from target 1. This illustrates the *interdependence of the associations* in a situation where a **persistent interference** (neighboring target) is present in addition to random interference (clutter).
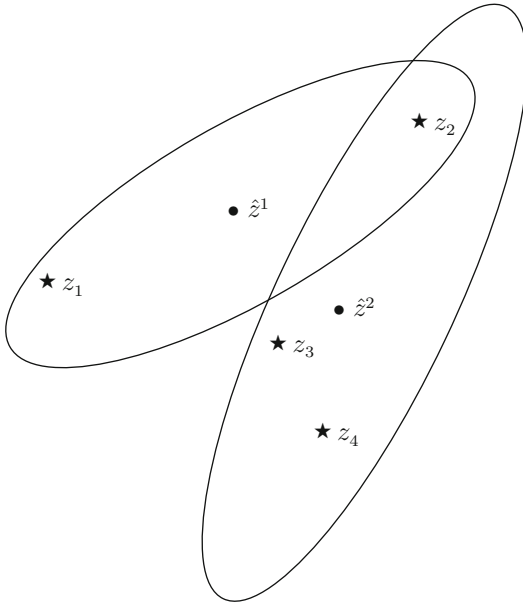
Data Association, Fig. 2 Two tracks with a measurement in the intersection of their validation regions

Up to this point, it was assumed that a measurement could have originated from *one of the targets* or from *clutter*. However, in view of the fact that any signal processing system has an inherent **finite resolution** capability, an additional possibility has to be considered:

   z_2 could be the result of the **merging** of the detections from the two targets – it is an **unresolved measurement**.

This constitutes a fourth origin hypothesis for a measurement that lies in the intersection of two validation regions. Most tracking algorithms ignore the possibility that a measurement is an unresolved one.

This illustrates only the difficulty of association of measurements to tracks *at one point in time*. The full problem, as will be discussed later, consists of associating measurements *across time*.

Approaches to Tracking and Data Association

The problem of **tracking and data association** is a **hybrid** problem because it is characterized by:

(1) Continuous uncertainties – state estimation in the presence of continuous noises
(2) Discrete uncertainties – which measurement(s) should be used in the estimation process

Assuming the goal is to obtain the MMSE estimate of the target state – its conditional mean – one can distinguish the following approaches.

Pure MMSE Approach
The **Pure MMSE Approach** to tracking and data association is obtained using the smoothing property of expectations (see, e.g., Bar-Shalom et al. 2001, Sect. 1.4.12), as follows:

$$\hat{x}^{\text{MMSE}} = E[x|Z] = E\{E[x|A, Z]|Z\}$$
$$= \sum_{A_i \in \mathcal{A}} E[x|A_i, Z]P\{A_i|Z\} \qquad (5)$$

where A is an association event (assuming a Bayesian model, with prior probabilities from which one can calculate posterior probabilities), and the summation is over all events A_i in the set $\mathcal{A}$ of mutually exclusive and exhaustive association events.

The above, which requires the evaluation of all the conditional (posterior) probabilities $P\{A_i|Z\}$, is a direct consequence of the **total probability theorem** (see, e.g., Bar-Shalom et al. 2001, Sect. 1.4.10), which yields the conditional pdf of the state as the following mixture

$$p(x|Z) = \sum_{A_i \in \mathcal{A}} p(x|A_i, Z)P\{A_i|Z\} \qquad (6)$$

In the linear-Gaussian case the above becomes a **Gaussian mixture**. Algorithms that fall in this category are PDAF and JPDAF, see Bar-Shalom et al. (2011).

MMSE-MAP Approach
The **MMSE-MAP Approach**, instead of enumerating and summing over all the association events, selects the one with highest posterior probability, namely,

$$A^{\text{MAP}} = \arg\max_i P\{A_i|Z\} \qquad (7)$$

and then

$$\hat{x}^{\text{MMSE-MAP}} = E[x|A^{\text{MAP}}, Z] \qquad (8)$$

The HOMHT as proposed by Reid (1979) falls in this category, see Bar-Shalom et al. (2011).

MMSE-ML Approach
The **MMSE-ML Approach** does not assume priors for the association events and relies on the maximum likelihood approach to select the event, that is,

$$A^{\text{ML}} = \arg\max_i p\{Z|A_i|\} \qquad (9)$$

and then

$$\hat{x}^{\text{MMSE-ML}} = E[x|A^{\text{ML}}, Z] \qquad (10)$$

The TOMHT falls into this category and S-D assignment (or MDA) is an implementation of this, see Bar-Shalom et al. (2011).

Heuristic Approaches
There are numerous simpler/heuristic approaches. The most common one relies on the distance metric (3) and makes the selection of which measurement is associated with which track based on the "nearest neighbor" rule. The same criterion can be used in a global cost function.

Remarks
It should be noted that the MMSE-MAP estimate (8) and the MMSE-ML estimate (10) are obtained assuming that the selected association is *correct* – a **hard decision**. This hard decision is sometimes correct, sometimes wrong. On the other hand, the pure MMSE estimate (5) yields a **soft decision** – it averages over all the possibilities. This soft decision is never totally correct, never totally wrong.

The uncertainties (covariances) associated with the MMSE-MAP and MMSE-ML estimates might be optimistic in view of the above observation. The uncertainty associated with the pure MMSE estimate will be *increased* (realistically) in view of the fact that it includes the data association uncertainty.

Estimation and Data Association in Nonlinear Stochastic Systems

The Model
Consider the discrete time stochastic system

$$\mathbf{x}(k + 1) = f[k, \mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)] \qquad (11)$$

where $\mathbf{x} \in \mathcal{R}^n$ is the stacked state vector of the targets under consideration, $\mathbf{u}(k)$ is a known input (included here for the sake of generality), and $\mathbf{v}(k)$ is the process noise with a known pdf. The measurements at time $k + 1$ are described by the stacked vector

$$\mathbf{z}(k+1) = h[k+1, \mathbf{x}(k+1), A(k+1), \mathbf{w}(k+1)] \qquad (12)$$

where $A(k + 1)$ is the data association event at $k + 1$ that specifies (i) which measurement component originated from which components of $\mathbf{x}(k + 1)$, namely, from which target, and (ii) which measurements are false, that is, originated from the clutter process. The vector $\mathbf{w}(k)$ is the observation noise, consisting of the error in the true measurement and the false measurements. The pdf of the false measurements and the probability mass function (pmf) of their number are also assumed to be known.

The noise sequences and false measurements are assumed to be white with known pdf and mutually independent. The initial state is assumed to have a known pdf and to be independent of the noises. Additional assumptions are given below for the optimal estimator, which evaluates the pdf of the state conditioned on the observations.

The optimal state estimator in the presence of data association uncertainty consists of the computation of the conditional pdf of the state $\mathbf{x}(k)$ given all the information available at time k, namely, the prior information about the initial state, the intervening inputs, and the sets of measurements through time k. The conditions under which the optimal state estimator consists of the computation of this pdf are presented in detail.

The Optimal Estimator for the Pure MMSE Approach

The **information set** available at k is

$$I^k = \{Z^k, U^{k-1}\} \tag{13}$$

where

$$Z^k \triangleq \{\mathbf{z}(j)\}_{j=1}^k \tag{14}$$

is the cumulative set of observations through time k, which subsumes the initial information Z^0, and U^{k-1} is the set of known inputs prior to time k.

For a stochastic system, an **information state** (Striebel 1965) is a function of the available information set that summarizes the past of the system in a probabilistic sense.

It can be shown that the conditional pdf of the state

$$p_k \triangleq p[\mathbf{x}(k)|I^k] \tag{15}$$

is an information state if (i) the two noise sequences (process and measurement) are white and mutually independent and (ii) the target detection and clutter/false measurement processes are white. Once the conditional pdf (15) is available, the **pure MMSE estimator**, i.e., the conditional mean, as well as the conditional variance, or covariance matrix, can be obtained.

The optimal estimator, which consists of the recursive functional relationship between the information states p_{k+1} and p_k, is given by

$$p_{k+1} = \psi[k+1, p_k, \mathbf{z}(k+1), \mathbf{u}(k)] \tag{16}$$

where

$$
\begin{aligned}
&\psi[k+1, p_k, \mathbf{z}(k+1), \mathbf{u}(k)] \\
&= \frac{1}{c} \sum_{i=1}^{M(k+1)} p[\mathbf{z}(k+1)|\mathbf{x}(k+1), A_i(k+1)] \\
&\quad \cdot \int p[\mathbf{x}(k+1)|\mathbf{x}(k), \mathbf{u}(k)] p_k \, d\mathbf{x}(k) \\
&\quad P\{A_i(k+1)\}
\end{aligned}
\tag{17}
$$

is the transformation that maps p_k into p_{k+1}; the integration in (17) is over the range of $\mathbf{x}(k)$ and c is the normalization constant.

The recursion (17) shows that the optimal MMSE estimator in the presence of data association uncertainty has the following properties:

P1. The pdf p_{k+1} is a weighted sum of pdfs, conditioned on the current time association events $A_i(k+1)$, $i = 1, \ldots, M(k+1)$, where $M(k+1)$ is the number of mutually exclusive and exhaustive association events at time $k+1$.

P2. If the exact previous pdf, which is the sufficient statistic, is available, then only the most recent association event probabilities are needed at each time.

However, the number of terms of the mixture in the right-hand side of (17) is, by time $k+1$, given by the product

$$M^{k+1} = \prod_{i=1}^{k+1} M(i) \tag{18}$$

which amounts to an exponential increase in time. This increase is similar to the increase in the number of the branches of the MHT hypothesis tree.

A detailed derivation of the recursion for the optimal estimator can be found in Bar-Shalom et al. (2011).

Track-to-Track Association

In addition to **measurement-to-track association (M2TA)**, an additional class of data association is **track-to-track association (T2TA)**. Following T2TA, **track-to-track fusion (T2TF)** may be performed to (hopefully) improve the overall tracking accuracy. For more details on track fusion, see Bar-Shalom et al. (2011).

It is desired first to test the hypothesis that two tracks pertain to the same target. The optimal test would require using the entire data base (the sequences of measurements that form the tracks) through the present time k and is not practical. In view of this, the test to be presented is based only on the most recent estimates from the tracks. The test based on the state estimates within a time window is discussed in Tian and Bar-Shalom (2009).

Association of Tracks with Independent Errors

Let $\hat{x}^i(k)$ be the estimated state of a target by sensor i with its own information processor. Assume that one has an estimate $\hat{x}^j(k)$ from sensor j, corresponding to the same time. Both can be current estimates or one can be a prediction as long as they pertain to the same time (the second time argument has been omitted for simplicity).

The corresponding covariances are denoted as $P^m(k), m = i, j$. The state estimation errors at different sensors (local trackers),

$$\tilde{x}^i(k) = x^i(k) - \hat{x}^i(k) \qquad (19)$$

$$\tilde{x}^j(k) = x^j(k) - \hat{x}^j(k) \qquad (20)$$

where x^i and x^j are the corresponding true states, are assumed to be *independent*. This is the **state estimation error independence assumption**.

Remark 2 As shown in the sequel, for *independent sensors*, the state estimation errors for the same target are *dependent* in the *presence of process noise*.

Denote the difference of the two estimates as

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k) - \hat{x}^j(k) \qquad (21)$$

This is the estimate of the difference of the true states

$$\Delta^{ij}(k) = x^i(k) - x^j(k) \qquad (22)$$

The **same target hypothesis** is that the true states are equal,

$$H_0: \qquad \Delta^{ij}(k) = 0 \qquad (23)$$

while the *different target* alternative is

$$H_1: \qquad \Delta^{ij}(k) \neq 0 \qquad (24)$$

While (21) is the appropriate statistic to test whether (22) is zero or not, the rigorous proof of this fact is presented in Bar-Shalom et al. (2011).

The error in the difference between the state estimates

$$\tilde{\Delta}^{ij}(k) = \Delta^{ij}(k) - \hat{\Delta}^{ij}(k) \qquad (25)$$

is zero mean and has covariance

$$T^{ij}(k) \triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\}$$
$$= E\{[\tilde{x}^i(k) - \tilde{x}^j(k)][\tilde{x}^i(k) - \tilde{x}^j(k)]'\} \qquad (26)$$

given, under the **error independence assumption**, by

$$T^{ij}(k) = P^i(k) + P^j(k) \qquad (27)$$

Assuming the estimation errors to be Gaussian, the test of H_0 vs. H_1 – the **T2TA** test – is

Accept H_0 if

$$D \triangleq \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \leq D_\alpha \quad (28)$$

The threshold D_α is such that

$$P\{D > D_\alpha | H_0\} = \alpha \qquad (29)$$

where, e.g., $\alpha = 0.01$. From the Gaussian assumption, the threshold is the $1 - \alpha$ point of the **chi-square distribution** with n_z degrees of freedom (Bar-Shalom et al. 2001)

$$D_\alpha = \chi^2_{n_z}(1 - \alpha) \qquad (30)$$

Association of Tracks with Dependent Errors

In the previous section, the association testing was done under the assumption that the estimation errors in these tracks are independent. However, as shown in Bar-Shalom et al. (2011), *whenever there is process noise* (or, in general, motion uncertainty), the track errors based on data from *independent sensors* are *dependent*.

The *dependence* between the estimation errors $\tilde{x}^i(k|k)$ and $\tilde{x}^j(k|k)$ from the two tracks arises from the *common process noise* which contributes to both errors. This is due to the fact that there is a common motion equation for both trackers.

The testing of the hypothesis that the two tracks under consideration originated from the same target is done in the same manner as before,

except for the following modification to account for the *dependence* of their state estimation errors.

The covariance associated with the difference of the estimates

$$\hat{\Delta}^{ij}(k) = \hat{x}^i(k|k) - \hat{x}^j(k|k) \qquad (31)$$

is, accounting for the dependence,

$$T^{ij}(k) \triangleq E\{\tilde{\Delta}^{ij}(k)\tilde{\Delta}^{ij}(k)'\}$$
$$= E\{[\tilde{x}^i(k|k) - \tilde{x}^j(k|k)][\tilde{x}^i(k|k)$$
$$- \tilde{x}^j(k|k)]'\} \qquad (32)$$

and, with the known cross-covariance P^{ij}, is given by the expression

$$T^{ij}(k) = P^i(k|k) + P^j(k|k)$$
$$- P^{ij}(k|k) - P^{ji}(k|k) \qquad (33)$$

Note the difference between the above and (27).

Effect of the Dependence

The effect of the dependence between the estimation errors is to *reduce* the covariance of the difference (31) of the estimates. This is due to the fact that the cross-covariance term reflects a positive correlation between the estimation errors (this is always the case for linear systems).

The Test

The hypothesis testing for the **track-to-track association** *with the dependence accounted for* is done in the same manner as before in (28), except that the "smaller" covariance from (33) is used in the test statistic, which is, as before, the **normalized distance squared** between the estimates

$$D = \hat{\Delta}^{ij}(k)'[T^{ij}(k)]^{-1}\hat{\Delta}^{ij}(k) \qquad (34)$$

The Cross-Covariance of the Estimation Errors

The **cross-covariance recursion** for *synchronized sensors* can be shown to be (see Bar-Shalom et al. 2011)

$$P^{ij}(k|k) \triangleq E[\tilde{x}^i(k|k)\tilde{x}^j(k|k)']$$
$$= [I - W^i(k)H^i(k)]$$
$$\cdot [F(k-1)P^{ij}(k-1|k-1)F(k-1)'$$
$$+ Q(k-1)][I - W^j(k)H^j(k)]' \qquad (35)$$

This is a *linear recursion* – a Lyapunov-type equation – and its initial condition is, assuming the initial errors to be uncorrelated,

$$P^{ij}(0|0) = 0 \qquad (36)$$

This is a reasonable assumption in view of the fact that the initial estimates are usually based on the initial measurements, which were assumed to have independent errors.

The cross-covariance for the case of asynchronous sensors can be found in Bar-Shalom et al. (2011).

Summary and Future Directions

This entry surveyed the issues involved in data association (specifically M2TA and T2TA) with regard to multitarget-multisensor tracking systems.

The future developments in this topic will be in regard to the use of new feature variables and classification in data association (some preliminary results are in Bar-Shalom et al. (2011)).

Cross-References

▶ Estimation for Random Sets
▶ Estimation, Survey on

Bibliography

Bar-Shalom Y, Li XR, Kirubarajan T (2001) Estimation with applications to tracking and navigation: theory, algorithms and software. Wiley, New York
Bar-Shalom Y, Willett PK, Tian X (2011) Tracking and data fusion. YBS, Storrs

Blackman SS, Popoli R (1999) Design and analysis of modern tracking systems. Artech House, Norwood

Mallick M, Krishnamurthy V, Vo BN (eds) (2013) Integrated tracking, classification, and sensor management. Wiley, Hoboken

Reid DB (1979) An algorithm for tracking multiple targets. IEEE Trans Autom Control 24:843–854

Striebel C (1965) Sufficient statistics in the optimum control of stochastic systems. J Math Anal Appl 12:576–592

Tian X, Bar-Shalom Y (2009) Track-to-track fusion configurations and association in a sliding window. J Adv Inf Fusion 4(2):146–164

Data Rate of Nonlinear Control Systems and Feedback Entropy

Christoph Kawan
Courant Institute of Mathematical Sciences, New York University, New York, USA

Abstract

Topological feedback entropy is a measure for the smallest information rate in a digital communication channel between the coder and the controller of a control system, above which the control task of rendering a subset of the state space invariant can be solved. It is defined purely in terms of the open-loop system without making reference to a particular coding and control scheme and can also be regarded as a measure for the inherent rate at which the system generates "invariance information."

Keywords

Communication constraints; Controlled invariance; Invariance entropy; Minimal data rates; Stabilization

Introduction

In the theory of networked control systems, the assumption of classical control theory that information can be transmitted within control loops instantaneously, lossless, and with arbitrary precision is no longer satisfied. Realistic mathematical models of many important real-world communication and control networks have to take into account general data rate constraints in the communication channels, time delays, partial loss of information, and variable network topologies. This raises the question about the smallest possible information rate above which a given control task can be solved. Though networked control systems can have a complicated topology, consisting of multiple sensors, controllers, and actuators, a first step towards understanding the problem of minimal data rates is to analyze the simplest possible network topology, consisting of one controller and one dynamical system connected by a digital channel with a certain rate in bits per unit time. There is a wealth of literature concerned with the problem of stabilizing a system under different assumptions about the specific coding and control scheme, in this context. However, with few exceptions, mainly linear systems (both deterministic and stochastic) have been considered. A comprehensive and detailed overview of this literature until 2007 can be found in the survey Nair et al. (2007). The first systematic approach to the problem of minimal data rates for set invariance and stabilization of (deterministic, nonlinear) control systems was presented in Nair et al. (2004), where the notion of *topological feedback entropy* was introduced. This quantity, defined in terms of the open-loop control system, is a measure for the smallest data rate a communication channel may have if the system is supposed to solve the control task of rendering a subset of the state space invariant. Other challenges that digital communication channels come along with are not yet taken into account here.

Feedback entropy was first introduced in Nair et al. (2004), using a similar approach via open covers as in the definition of topological entropy of for classical dynamical systems in Adler et al. (1965). In Colonius and Kawan (2009), a quantity named *invariance entropy* was defined which later turned out to be equivalent to the feedback entropy of Nair et al. (cf. Colonius et al. 2013). The notion of invariance entropy has been further

studied in the papers Kawan (2011a), Kawan (2011b), and Kawan (2011c). Several variations and generalizations have been introduced in Colonius (2010), Colonius (2012), Colonius and Kawan (2011), Da Silva (2013), and Hagihara and Nair (2013). The research monograph Kawan (2013) provides a comprehensive presentation of the results obtained so far in the deterministic case.

Definition

Topological feedback entropy is a nonnegative real-valued quantity which serves as a measure for the smallest possible data rate in a digital channel, connecting a coder to a controller, above which the controller is able to generate inputs which guarantee invariance of a given subset of the state space. In the literature, one finds several slightly differing versions. The original definition given in Nair et al. (2004) is (with minor modifications) as follows. Consider a discrete-time control system

$$x_{k+1} = F(x_k, u_k) = F_{u_k}(x_k), \qquad k \geq 0,$$

with $F : X \times U \to X$, where X is a topological space and U a nonempty set such that $F_u : X \to X$ is continuous for every $u \in U$. The transition map associated to this system is

$$\varphi : \mathbb{N}_0 \times X \times U^{\mathbb{N}_0} \to X,$$

$$\varphi(k, x, (u_n)) := F_{u_{k-1}} \circ \cdots \circ F_{u_1} \circ F_{u_0}(x).$$

A compact subset $K \subset X$ with nonempty interior is *(strongly) controlled invariant* if for every $x \in K$ there is an input $u \in U$ such that $F_u(x) \in$ int K. A triple $(\mathcal{A}, \tau, G)$ is called an *invariant open cover* of K if $\mathcal{A}$ is an open cover of K, τ is a positive integer, and $G : \mathcal{A} \to U^\tau$ is a map with components $G_0, G_1, \ldots, G_{\tau-1}$ which assign control values to all sets in $\mathcal{A}$ such that for every $A \in \mathcal{A}$ the finite sequence of controls $G(A)$ yields $\varphi(k, A, G(A)) \subset$ int K for $k = 1, 2, \ldots, \tau$. The *entropy of* $(\mathcal{A}, \tau, G)$ is defined as follows. For every sequence $\alpha = (A_i)_{i \geq 0}$ of sets in $\mathcal{A}$ one defines an associated sequence of controls by

$$\underline{u}(\alpha) = (u_0, u_1, u_2, \ldots) \quad \text{with } (u_l)_{l=(i-1)\tau}^{i\tau-1}$$
$$= G(A_{i-1}) \quad \text{for all } i \geq 0,$$

and for every $j \geq 1$ a set

$$B_j(\alpha) := \{x \in X \ : \ \varphi(i\tau, x, \underline{u}(\alpha)) \in A_i$$
$$\text{for } i = 0, 1, \ldots, j - 1\}.$$

The family $\mathcal{B}_j := \{B_j(\alpha) : \alpha \in \mathcal{A}^{\mathbb{N}_0}\}$ is an open cover of K. Letting $N(\mathcal{B}_j | K)$ denote the minimal cardinality of a finite subcover, the entropy of $(\mathcal{A}, \tau, G)$ is

$$h(\mathcal{A}, \tau, G) := \lim_{j \to \infty} \frac{1}{j\tau} \log_2 N(\mathcal{B}_j | K)$$
$$= \inf_{j \geq 1} \frac{1}{j\tau} \log_2 N(\mathcal{B}_j | K).$$

Finally, the topological feedback entropy (TFE) of K is given by

$$h_{\mathrm{fb}}(K) := \inf_{(\mathcal{A}, \tau, G)} h(\mathcal{A}, \tau, G),$$

where the infimum is taken over all invariant open covers of K.

A conceptually simpler but equivalent definition, introduced in Colonius and Kawan (2009), is the following. A subset $\mathcal{S} \subset U^\tau$ is called (τ, K)-*spanning* if for every $x \in K$ there is $\underline{u} \in \mathcal{S}$ with $\varphi(k, x, \underline{u}) \in$ int K for $k = 1, \ldots, \tau$. Writing $r_{\mathrm{inv}}(\tau, K)$ for the minimal cardinality of such a set, it can be shown that

$$h_{\mathrm{fb}}(K) = \lim_{\tau \to \infty} \frac{1}{\tau} \log_2 r_{\mathrm{inv}}(\tau, K)$$
$$= \inf_{\tau \geq 1} \frac{1}{\tau} \log_2 r_{\mathrm{inv}}(\tau, K).$$

This definition and several variations of it are mostly referred to by the name *invariance entropy* instead of feedback entropy. The intuition behind this definition is that a controller which receives a certain amount of information about the state, say n bits, can generate at most 2^n different control sequences to steer the system on a finite time interval and hence, the number of control sequences needed to

accomplish the control task on this interval is a measure for the necessary amount of information.

The different variations of feedback or invariance entropy which can be found in the literature are briefly summarized as follows. For simplicity, in this entry we refer to several of these variations by the name *(topological) feedback entropy*:

(i) Instead of requiring that trajectories enter the interior of K after one step of time, one can allow for a waiting time τ_0 before entering $\text{int} K$.

(ii) One can require that trajectories stay in K instead of $\text{int} K$ or that they stay in an arbitrarily small neighborhood of K, respectively.

(iii) One can restrict the set of initial states to a subset of $K' \subset K$. In this case, a set $\mathcal{S}$ of control sequences is (τ, K', K)-spanning if for every $x \in K'$ there is $\underline{u} \in \mathcal{S}$ with $\varphi(k, x, \underline{u}) \in \text{int} K$ for $k = 1, \ldots, \tau$.

(iv) Feedback entropy can be defined for other classes of systems, e.g., continuous-time deterministic systems, random control systems, or systems with piecewise continuous right-hand sides.

There is also a local version of topological feedback entropy (LTFE) which measures the smallest data rate for local uniform asymptotic stabilization at an equilibrium. Also for other control tasks there have been attempts to define corresponding versions of feedback or invariance entropy.

Comparison to Topological Entropy

Though there are similarities in the definitions of TFE and topological entropy of dynamical systems, which also reflect in similar properties, there is no direct relation between these two quantities. Topological entropy detects exponential complexity in the orbit structure of a dynamical system. In contrast, TFE measures the complexity of the control task to keep a system in a subset of the state space by applying appropriate inputs. If no escape from this subset is possible, the TFE is zero, no matter how complicated the orbit structure is. Hence, topological entropy is sensitive to the local behavior of the system,

while TFE in general is not. Interpreted in terms of information rates, topological entropy is a measure for the largest average rate of information about the initial state a dynamical system can generate. TFE measures the smallest rate of information about the state of the system above which a controller is able to render the set invariant. It should also be mentioned that topological entropy was first introduced as a topological counterpart of the measure-theoretic entropy defined by Kolmogorov and Sinai, and that the two notions are related by the variational principle, which asserts that the topological entropy is the supremum of the measure-theoretic entropies with respect to all invariant probability measures of the given system. For TFE, so far no convincing measure-theoretic approach exists. An excellent survey on the entropy theory of dynamical systems can be found in Katok (2007).

The Data Rate Theorem

The *data rate theorem* for the TFE confirms that the infimal data rate in a coding and control loop which guarantees strong controlled invariance of a set K is equal to $h_{\text{fb}}(K)$. More precisely, suppose that a sensor which measures the state of the system at discrete times $\tau_k = k\tau, k = 0, 1, 2, \ldots$, is connected to a coder which at time τ_k has a finite alphabet S_k of symbols available. The measured state is coded by use of this alphabet and the corresponding symbol is sent via a noiseless digital channel to a controller which generates an input sequence of length τ. This sequence is used to steer the system until the next symbol arrives at time τ_{k+1}. The associated *asymptotic average bit rate*, which depends on the sequence $S = (S_k)_{k \geq 0}$ of coding alphabets, is given by

$$R(S) = \lim_{k \to \infty} \frac{1}{k\tau} \sum_{i=0}^{k-1} \log_2 |S_i|.$$

If the limit does not exist, one may replace it with $\liminf$ or $\limsup$. The data rate theorem establishes the equality

$$h_{\text{fb}}(K) = \inf_S R(S),$$

where the infimum is taken over all coding and control loops which guarantee strong controlled invariance of K, i.e., for initial states in K they generate trajectories $(x_k)_{k \geq 0}$ with $x_k \in \text{int} K$ for $k = 1, 2, \ldots$. Similar data rate theorems can be proved for other variants of feedback entropy. In particular, the data rate theorem for the LTFE asserts that the infimal bit rate for local uniform asymptotic stabilization at an equilibrium is given by the LTFE. Proofs of different data rate theorems can be found in Nair et al. (2004), Hagihara and Nair (2013), and Kawan (2013).

Estimates and Formulas

Linear Systems

For linear systems, under reasonable assumptions, the feedback entropy is given by the sum of the unstable eigenvalues of the dynamical matrix, i.e., if the system is given by $x_{k+1} = Ax_k + Bu_k$, then

$$h_{\text{fb}}(K) = \sum_{\lambda \in \text{Sp}(A)} \max \left\{ 0, n_\lambda \log_2 |\lambda| \right\}, \quad (1)$$

where $\text{Sp}(A)$ denotes the spectrum of A and n_λ is the algebraic multiplicity of the eigenvalue λ (cf. Colonius and Kawan 2009). It is worth to mention that the TFE therefore coincides with the topological entropy of the uncontrolled system $x_{k+1} = Ax_k$, as defined by Bowen for maps on non-compact metric spaces (cf. Bowen 1971). However, this is a special property of linear systems and is related to the facts that (i) there is no difference between the local and the global dynamical behavior of uncontrolled linear systems and that (ii) the control sequence does not affect the exponential complexity of the dynamics, since it only appears as an additive term in the transition map of the system. Formula (1) is in correspondence with several former results on minimal data rates for stabilization of linear systems. Thinking of the definition of feedback entropy via spanning sets of control sequences, its interpretation is that in order to guarantee invariance of a bounded set, the only reason for exponential growth of the number of necessary

inputs as time increases is the volume expansion of the open-loop system in the unstable subspace.

Upper Bounds Under Controllability Assumptions

If the state space of the control system is a differentiable manifold and the right-hand side is continuously differentiable, under certain controllability assumptions upper bounds for the feedback entropy can be formulated in terms of the Lyapunov exponents of periodic trajectories (for the concept of Lyapunov exponents, see, e.g., Barreira and Valls 2008) (cf. Kawan 2011b, 2013; Nair et al. 2004). More precisely, if there is a periodic trajectory in the interior of the given set K such that the linearization along this trajectory is controllable, and if complete approximate controllability holds on the interior of K (cf. Colonius and Kliemann 2000), then

$$h_{\text{fb}}(K) \leq \sum_\lambda \max\{0, n_\lambda \lambda\}, \quad (2)$$

where the sum is taken over all Lyapunov exponents λ of the periodic trajectory and n_λ denotes the multiplicity of λ. Using the definition of feedback entropy in terms of (τ, K)-spanning sets, one can prove this by constructing spanning sets of control functions which first steer all initial states in K into a small neighborhood of a point on the given periodic orbit and then, by use of local controllability, keep the corresponding trajectories in a neighborhood of the periodic trajectory for arbitrary future times. Similar ideas first have been used in Nair et al. (2004) to prove that the LTFE at an equilibrium is given by the sum of the unstable eigenvalues of the linearization about this equilibrium. For systems given by differential equations the upper estimate (2) can be improved under additional regularity assumptions. Assuming that the system is smooth and satisfies the strong jet accessibility rank condition (cf. Coron 1994), one can show that both assumptions, controllability of the linearization and periodicity, can be omitted. The only restriction that remains is that the trajectory must not leave a compact subset of the interior of K. However, in the case of nonperiodic trajectories, the sum

of the positive Lyapunov exponents has to be replaced by the maximal Lyapunov exponent of the induced linear flow on the exterior bundle of the manifold. For control-affine systems, strong jet accessibility can be weakened to local accessibility. In general, it is unlikely that such upper bounds are tight, since they are related to very specific control strategies for making the given set invariant.

Lower Bounds, Volume Growth Rates, and Escape Rates

A general approach to obtain lower bounds of feedback entropy is via a volume growth argument, which in its simplest form works as follows. Every (τ, K)-spanning set $\mathcal{S}$ defines a cover of K, consisting of the sets (cf. Kawan 2011a,c)

$$K_{\tau,\underline{u}} = \{x \in K \ : \ \varphi(k, x, \underline{u}) \in \mathrm{int}K,$$
$$1 \le k \le \tau\}, \qquad \underline{u} \in \mathcal{S}.$$

It follows that $\varphi_{\tau,\underline{u}}(K_{\tau,\underline{u}}) \subset K$ and hence, since K is bounded, the volume expansion under $\varphi_{\tau,\underline{u}} = \varphi(\tau, \cdot, \underline{u})$ gives upper bounds for the volumes of the sets $K_{\tau,\underline{u}}$, which result in a lower bound for the number of these sets. For instance, the lower estimate in (1) can be established by applying this argument to the system which arises by projection of the given linear system to the unstable subspace of the uncontrolled part $x_{k+1} = Ax_k$. A refinement of this idea also leads to lower estimates of feedback entropy for inhomogeneous bilinear systems in terms of volume growth rates or Lyapunov exponents on unstable bundles, respectively. For nonlinear systems, in general only very rough estimates can be obtained by this method. However, a variation of the volume growth argument leads to a lower bound of the form

$$h_{\mathrm{fb}}(K) \ge -\liminf_{\tau \to \infty} \frac{1}{\tau} \log \sup_{\underline{u}} \mu(K_{\tau,\underline{u}}),$$

where μ denotes a reference measure on the state space. The right-hand side of this inequality can be considered as a uniform *escape rate* from the set K, which under sufficiently strong hyperbolicity assumptions can be estimated in terms of other quantities such as Lyapunov exponents and dynamical entropies. Key references for escape rates in the classical theory of dynamical systems are (Young (1990) and Demers and Young (2006)).

Summary and Future Directions

The theory of feedback entropy for finite-dimensional deterministic systems is very far from being complete. The currently available results only give valuable information in very regular situations, and even those are not fully understood. For the further development of this theory, it will be necessary to combine control-theoretic methods with techniques from different fields such as classical, random, and nonautonomous dynamical systems. Some of the main focuses of future research will probably be the following:

- The generalization of feedback entropy to more complex network topologies
- The formulation of a feedback entropy theory for stochastic systems
- The development of a probabilistic (resp. measure-theoretic) version of feedback entropy for both deterministic and stochastic systems, which is related to the topological version via a variational principle
- The numerical computation of feedback entropy

Cross-References

▶ Quantized Control and Data Rate Constraints

Bibliography

Adler RL, Konheim AG, McAndrew MH (1965) Topological entropy. Trans Amer Math Soc 114:309–319

Barreira L, Valls C (2008) Stability of nonautonomous differential equations. Lecture notes in mathematics, vol. 1926. Springer, Berlin

Bowen R (1971) Entropy for group endomorphisms and homogeneous spaces. Trans Am Math Soc 153:401–414

Colonius F (2010) Minimal data rates and invariance entropy. Electronic Proceedings of the conference on mathematical theory of networks and systems (MTNS), Budapest, 5–9 July 2010

Colonius F (2012) Minimal bit rates and entropy for stabilization. SIAM J Control Optim 50: 2988–3010

Colonius F, Kawan C (2009) Invariance entropy for control systems. SIAM J Control Optim 48: 1701–1721

Colonius F, Kawan C (2011) Invariance entropy for outputs. Math Control Signals Syst 22: 203–227

Colonius F, Kliemann W (2000) The dynamics of control. Birkhäuser-Verlag, Boston

Colonius F, Kawan C, Nair GN (2013) A note on topological feedback entropy and invariance entropy. Syst Control Lett 62:377–381

Coron J-M (1994) Linearized control systems and applications to smooth stabilization. SIAM J Control Optim 32:358–386

Da Silva AJ (2013) Invariance entropy for random control systems. Math Control Signals Syst 25: 491–516

Demers MF, Young L-S (2006) Escape rates and conditionally invariant measures. Nonlinearity 19:377–397

Hagihara R, Nair GN (2013) Two extensions of topological feedback entropy. Math Control Signals Syst 25:473–490

Katok A (2007) Fifty years of entropy in dynamics: 1958–2007. J Mod Dyn 1:545–596

Kawan C (2011a) Upper and lower estimates for invariance entropy. Discret Contin Dyn Syst 30:169–186

Kawan C (2011b) Invariance entropy of control sets. SIAM J Control Optim 49:732–751

Kawan C (2011c) Lower bounds for the strict invariance entropy. Nonlinearity 24:1910–1935

Kawan C (2013) Invariance entropy for deterministic control systems – an introduction. Lecture notes in mathematics vol 2089. Springer, Berlin

Nair GN, Evans RJ, Mareels IMY, Moran W (2004) Topological feedback entropy and nonlinear stabilization. IEEE Trans Autom Control 49:1585–1597

Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. Proc IEEE 95:108–137

Young L-S (1990) Large deviations in dynamical systems. Trans Am Math Soc 318:525–543

DES

Deterministic Description of Biochemical Networks

Jörg Stelling and Hans-Michael Kaltenbach
ETH Zürich, Basel, Switzerland

Abstract

Mathematical models of living systems are often based on formal representations of the underlying reaction networks. Here, we present the basic concepts for the deterministic nonspatial treatment of such networks. We describe the most prominent approaches for steady-state and dynamic analysis using systems of ordinary differential equations.

Keywords

Michaelis-Menten kinetics; Reaction networks; Stoichiometry

Introduction

A biochemical network describes the interconversion of biochemical species such as proteins or metabolites by chemical reactions. Such networks are ubiquitous in living cells, where they are involved in a variety of cellular functions such as conversion of metabolites into energy or building material of the cell, detection and processing of external and internal signals of nutrient availability or environmental stress, and regulation of genetic programs for development.

Reaction Networks

A biochemical network can be modeled as a dynamic system with the chemical concentration of each species taken as the states and dynamics described by the changes in species concentrations as they are converted by reactions. Assuming that species are homogeneously distributed in the reaction volume and that copy numbers are sufficiently high, we may ignore spatial and stochastic

effects and derive a system of ordinary differential equations (ODEs) to model the dynamics.

Formally, a biochemical network is given by r reactions $R_1, \ldots, R_r$ acting on n different chemical species $S_1, \ldots, S_n$. Reaction R_j is given by

$$R_j : \ \alpha_{1,j} S_1 + \cdots + \alpha_{n,j} S_n \longrightarrow \beta_{1,j} S_1$$
$$+ \cdots + \beta_{n,j} S_n \ ,$$
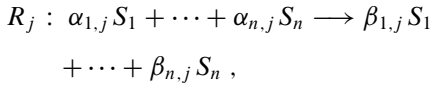
where $\alpha_{i,j}, \beta_{i,j} \in \mathbb{N}$ are called the *molecularities* of the species in the reaction. Their differences form the *stoichiometric matrix* $\mathbf{N} = (\beta_{i,j} - \alpha_{i,j})_{i=1\ldots n, j=1\ldots r}$, with $N_{i,j}$ describing the net effect of one turnover of R_j on the copy number of species S_i. The jth column is also called the *stoichiometry* of reaction R_j. The system can be opened to an un-modeled environment by introducing inflow reactions $\emptyset \rightarrow S_i$ and outflow reactions $S_i \rightarrow \emptyset$.

For example, consider the following reaction network:

$$R_1 : \qquad E + S \rightarrow E \cdot S$$
$$R_2 : \qquad E \cdot S \rightarrow E + S$$
$$R_3 : \qquad E \cdot S \rightarrow E + P$$
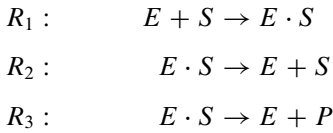
Here, an enzyme E (a protein that acts as a catalyst for biochemical reactions) binds to a substrate species S, forming an intermediate complex $E \cdot S$ and subsequently converting S into a product P. Note that the enzyme-substrate binding is reversible, while the conversion to a product is irreversible. This network contains $r = 3$ reactions interconverting $n = 4$ chemical species $S_1 = S$, $S_2 = P$, $S_3 = E$, and $S_4 = E \cdot S$. The stoichiometric matrix is

$$\mathbf{N} = \begin{pmatrix} -1 & +1 & 0 \\ 0 & 0 & +1 \\ -1 & +1 & +1 \\ +1 & -1 & -1 \end{pmatrix} \ .$$

Dynamics

Let $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))^T$ be the vector of concentrations, that is, $x_i(t)$ is the concentration of S_i at time t. Abbreviating this state vector as $\mathbf{x}$

by dropping the explicit dependence on time, its dynamics is governed by a system of n ordinary differential equations:

$$\frac{d}{dt}\mathbf{x} \ = \ \mathbf{N} \cdot \mathbf{v}(\mathbf{x}, \mathbf{p}) \ . \tag{1}$$

Here, the *reaction rate* vector $\mathbf{v}(\mathbf{x}, \mathbf{p}) = (v_1(\mathbf{x}, \mathbf{p}_1), \ldots, v_r(\mathbf{x}, \mathbf{p}_r))^T$ gives the rate of conversion of each reaction per unit-time as a function of the current system state and of a set of parameters $\mathbf{p}$.

A typical reaction rate is given by the *mass-action rate law*

$$v_j(\mathbf{x}, \mathbf{p}_j) = k_j \cdot \prod_{i=1}^{n} x_i^{\alpha_{i,j}} \ ,$$

where the rate constant $k_j > 0$ is the only parameter and the rate is proportional to the concentration of each species participating as an educt (consumed component) in the respective reaction.

Equation (1) decomposes the system into a time-independent and linear part described solely by the topology and stoichiometry of the reaction network via $\mathbf{N}$ and a dynamic and typically nonlinear part given by the reaction rate laws $\mathbf{v}(\cdot, \cdot)$. One can define a directed graph of the network with one vertex per state and take $\mathbf{N}$ as the (weighted) incidence matrix. Reaction rates are then properties of the resulting edges. In essence, the equation describes the change of each species' concentration as the sum of the current reaction rates. Each rate is weighted by the molecularity of the species in the corresponding reaction; it is negative if the species is consumed by the reaction and positive if it is produced.

Using mass-action kinetics throughout, and using the species name instead of the numeric subscript, the reaction rates and parameters of the example network are given by

$$v_1(\mathbf{x}, \mathbf{p}_1) = k_1 \cdot x_E(t) \cdot x_S(t)$$
$$v_2(\mathbf{x}, \mathbf{p}_2) = k_2 \cdot x_{E \cdot S}(t)$$
$$v_3(\mathbf{x}, \mathbf{p}_3) = k_3 \cdot x_{E \cdot S}(t)$$
$$\mathbf{p} = (k_1, k_2, k_3)$$

The system equations are then

$$\frac{d}{dt}x_S = -k_1 \cdot x_S \cdot x_E + k_2 \cdot x_{E \cdot S}$$

$$\frac{d}{dt}x_P = k_3 \cdot x_{E \cdot S}$$

$$\frac{d}{dt}x_E = -k_1 \cdot x_S \cdot x_E + k_2 \cdot x_{E \cdot S} + k_3 \cdot x_{E \cdot S}$$

$$\frac{d}{dt}x_{E \cdot S} = k_1 \cdot x_S \cdot x_E - k_2 \cdot x_{E \cdot S} - k_3 \cdot x_{E \cdot S} \ .$$

Steady-State Analysis

A reaction network is in *steady state* if the production and consumption of each species are balanced. Steady-state concentrations $\mathbf{x}^*$ then satisfy the equation

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v}(\mathbf{x}^*, \mathbf{p}) \ .$$

Computing steady-state concentrations requires explicit knowledge of reaction rates and their parameter values. For biochemical reaction networks, these are often very difficult to obtain. An alternative is the computation of *steady-state fluxes* $\mathbf{v}^*$, which only requires solving the system of homogeneous linear equations

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v} \ . \tag{2}$$

Lower and upper bounds v_i^l, v_i^u for each flux v_i can be given such that $v_i^l \leq v_i \leq v_i^u$; an example is an irreversible reaction R_i which implies $v_i^l = 0$. The set of all feasible solutions then forms a pointed, convex, polyhedral *flux cone* in $\mathbb{R}^r$. The rays spanning the flux cone correspond to *elementary flux modes (EFMs)* or *extreme pathways (EPs)*, minimal subnetworks that are already balanced. Each feasible steady-state flux can be written as a nonnegative combination

$$\mathbf{v}^* = \sum_i \lambda_i \cdot \mathbf{e}_i, \quad \lambda_i \geq 0$$

of EFMs $\mathbf{e}_1, \mathbf{e}_2, \ldots,$ where the λ_i are the corresponding weights.

Even if in steady state, living cells grow and divide. Growth of a cell is often described by a combination of fluxes $\mathbf{b}^T \cdot \mathbf{v}$, the total production rate of relevant metabolites to form new biomass. The *biomass function* given by $\mathbf{b} \in \mathbb{R}^r$ is determined experimentally. The technique of *flux balance analysis (FBA)* then solves the linear program

$$\max_{\mathbf{v}} \ \mathbf{b}^T \cdot \mathbf{v}$$

subject to

$$\mathbf{0} = \mathbf{N} \cdot \mathbf{v}$$

$$v_i^l \leq v_i \leq v_i^u$$

to yield a feasible flux vector that balances the network while maximizing growth. Alternative objective functions have been proposed, for instance, for higher organisms that do not necessarily maximize the growth of each cell.

Quasi-Steady-State Analysis

In many reaction mechanisms, a *quasi-steady-state assumption (QSSA)* can be made, postulating that the concentration of some of the involved species does not change. This assumption is often justified if reaction rates differ hugely, leading to a time scale separation, or if some concentrations are very high, such that their change is negligible for the mechanism. A typical example is the derivation of *Michaelis-Menten kinetics*, which corresponds to our example network. There, we may assume that the concentration of the intermediate species $E \cdot S$ stays approximately constant on the time scale of the overall conversion of substrate into product and that the substrate, at least initially, is in much larger abundance than the enzyme. On the slower time scale, this leads to the *Michaelis-Menten rate law*:

$$v_P = \frac{d}{dt}x_P(t) = \frac{v_{\max} \cdot x_S(t)}{K_m + x_S(t)} \ ,$$

with a maximal rate $v_{\max} = k_3 \cdot x_E^{\text{tot}}$, where x_E^{tot} is the total amount of enzyme and the Michaelis-Menten constant $K_m = (k_2 + k_3)/k_1$ as a direct relation between substrate concentration and production rate. This approximation reduces the number of states by two. Both parameters

of the Michaelis-Menten rate law are also better suited for experimental determination: v_{max} is the highest rate achievable and K_m corresponds to the substrate concentration that yields a rate of $v_{max}/2$.

Cooperativity and Ultra-sensitivity

In the Michaelis-Menten mechanism, the production rate gradually increases with increasing substrate concentration, until saturation (Fig. 1; $h = 1$). A different behavior is achieved if the enzyme has several binding sites for the substrate and these sites interact such that occupation of one site alters the affinity of the other sites positively or negatively, phenomena known as positive and negative cooperativity, respectively. With QSSA arguments as before, the fraction of enzymes completely occupied by substrate molecules at time t is given by

$$ v = \frac{v_{max} \cdot x_S(t)}{K^h + x_S^h(t)} $$

where $K > 0$ is a constant and $h > 0$ is the *Hill coefficient*. The Hill coefficient determines the shape of the response with increasing substrate concentration: a coefficient of $h > 1$ ($h < 1$) indicates positive (negative) cooperativity; $h = 1$ reduces to the Michaelis-Menten mechanism. With increasing coefficient h, the response changes from gradual to switch-like, such that the transition from low to high response becomes more rapid as indicated in Fig. 1. This phenomenon is also known as *ultra-sensitivity*.

Constrained Dynamics

Due to the particular structure of the system equation (1), the trajectories $\mathbf{x}(t)$ of the network with $\mathbf{x}_0 = \mathbf{x}(0)$ are confined to the *stoichiometric subspace*, the intersection of $\mathbf{x}_0 + \text{Img}\mathbf{N}$ with the positive orthant. *Conservation relations* that describe conservation of mass are thus found as solutions to

$$ \mathbf{c}^T \cdot \mathbf{N} = \mathbf{0} \;, $$

and two initial conditions $\mathbf{x}_0, \mathbf{x}_0'$ lead to the same stoichiometric subspace if $\mathbf{c}^T \cdot \mathbf{x}_0 = \mathbf{c}^T \cdot \mathbf{x}_0'$. This allows for the analysis of, for example, bistability using only the reaction network structure.

Summary and Future Directions

Reactions networks, even in simple cells, typically encompass thousands of components and reactions, resulting in potentially high-dimensional nonlinear dynamic systems. In contrast to engineered systems, biology is characterized by a high degree of uncertainty of both model structure and parameter values. Therefore, system identification is a central problem in this domain. Specifically, advanced methods for model topology and parameter identification as well as for uncertainty quantification need to be developed that take into account the very limited observability of biological systems. In addition, biological systems operate on multiple time, length, and concentration scales. For example, genetic regulation usually operates on the time scale of minutes and involves very few molecules, whereas metabolism is significantly faster and states are well approximated by species

Deterministic Description of Biochemical Networks, Fig. 1 Responses for cooperative enzyme reaction with Hill coefficient $h = 1, 3, 10$, respectively. All other parameters are set to 1
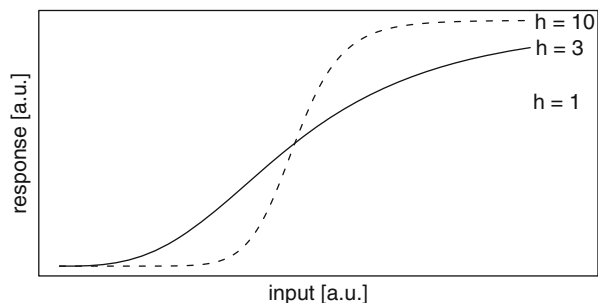
concentrations. Corresponding systematic frameworks for multiscale modeling, however, are currently lacking.

Cross-References

Bibliography

Craciun G, Tang Y, Feinberg M (2006) Understanding bistability in complex enzyme-driven reaction networks. Proc Natl Acad Sci USA 103(23):8697–8702

Higham DJ (2008) Modeling and simulating chemical reactions. SIAM Rev 50(2):347–368

LeDuc PR, Messner WC, Wikswo JP (2011) How do control-based approaches enter into biology? Annu Rev Biomed Eng 13:369–396

Sontag E (2005) Molecular systems biology and control. Eur J Control 11(4–5):396–435

Szallasi Z, Stelling J, Periwal V (eds) (2010) System modeling in cellular biology: from concepts to nuts and bolts. MIT, Cambridge

Tyson JJ, Chen KC, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol 15(2):221–231

Diagnosis of Discrete Event Systems

Stéphane Lafortune
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA

Abstract

We discuss the problem of event diagnosis in partially observed discrete event systems. The objective is to infer the past occurrence, if any, of an unobservable event of interest based on the observed system behavior and the complete model of the system. Event diagnosis is performed by diagnosers that are synthesized from the system model and that observe the system behavior at run-time. Diagnosability analysis is the off-line task of determining which events of interest can be diagnosed at run-time by diagnosers.

Keywords

Diagnosability; Diagnoser; Fault diagnosis; Verifier

Introduction

In this entry, we consider discrete event systems that are partially observable and discuss the two related problems of event diagnosis and diagnosability analysis. Let the DES of interest be denoted by M with event set E. Since M is partially observable, its set of events E is the disjoint union of a set of *observable events*, denoted by E_o, with a set of *unobservable* events, denoted by E_{uo}: $E = E_o \cup E_{uo}$. At this point, we do not specify how M is represented; it could be an automaton or a Petri net. Let L_M be the set of all strings of events in E that the DES M can execute, i.e., the (untimed) language model of the system; cf. the related entries, ▶ Models for Discrete Event Systems: An Overview, ▶ Supervisory Control of Discrete-Event Systems, and ▶ Modeling, Analysis, and Control with Petri Nets. The set E_{uo} captures the fact that the set of sensors attached to the DES M is limited and may not cover all the events of the system. Unobservable events can be internal system events that are not directly "seen" by the monitoring agent that observes the behavior of M for diagnosis purposes. They can also be fault events that are included in the system model but are not directly observable by a dedicated sensor. For the purpose of diagnosis, let us designate a specific unobservable event of interest and denote it by $d \in E_{uo}$. Event d could be a fault event or some other significant event that is unobservable.

Before we can state the problem of event diagnosis, we need to introduce some notation. E^* is the set of all strings of any length $n \in \mathbb{N}$ that can be formed by concatenating elements of E. The unique string of length $n = 0$ is denoted by ε and is the identity element of concatenation. As in article ▶ Supervisory Control of Discrete-Event Systems, section "Supervisory Control Under Partial Observations," we define the projection function $P : E^* \rightarrow E_o^*$ that "erases" the unobservable events in a string and replaces them by ε. The function P is naturally extended to a set of strings by applying it to each string in the set, resulting in a set of projected strings. The observed behavior of M is the language $P(L_M)$ over event set E_o.

The problem of *event diagnosis*, or simply *diagnosis*, is stated as follows: how to infer the past occurrence of event d when observing strings in $P(L_M)$ at run-time, i.e., during the operation of the system? This is model-based inferencing, i.e., the monitoring agent knows L_M and the partition $E = E_o \cup E_{uo}$, and it observes strings in $P(L_M)$. When there are multiple events of interest, d_1 to d_n, and these events are *fault* events, we have a problem of *fault diagnosis*. In this case, the objective is not only to determine that a fault has occurred (commonly referred to as "fault detection") but also to identify which fault has occurred, namely, which event d_i (commonly referred to as "fault isolation and identification"). Fault diagnosis requires that L_M contains not only the nominal or fault-free behavior of the system but also its behavior after the occurrence of each fault event d_i of interest, i.e., its faulty behavior. Event d_i is typically a fault of a component that leads to degraded behavior on the part of the system. It is not a catastrophic failure that would cause the system to completely stop operating, as such a failure would be immediately observable. The decision on which fault events d_i, along with their associated faulty behaviors, to include in the complete model L_M is a design one that is based on practical considerations related to the diagnosis objectives.

A complementary problem to event diagnosis is that of *diagnosability analysis*. Diagnosability analysis is the off-line task of determining, on the basis of L_M and of E_o and E_{uo}, if any and all occurrences of the given event of interest d will eventually be diagnosed by the monitoring agent that observes the system behavior.

Event diagnosis and diagnosability analysis arise in numerous applications of systems that are modeled as DES. We mention a few application areas where DES diagnosis theory has been employed. In heating, ventilation, and air-conditioning systems, components such as valves, pumps, and controllers can fail in degraded modes of operation, such as a valve gets stuck open or stuck closed or a pump or controller gets stuck on or stuck off. The available sensors may not directly observe these faults, as the sensing abilities are limited. Fault diagnosis techniques are essential, since the components of the system are often not easily accessible. In monitoring communication networks, faults of certain transmitters or receivers are not directly observable and must be inferred from the set of successful communications and the topology of the network. In document processing systems, faults of internal components can lead to jams in the paper path or a decrease in image quality, and while the paper jam or the image quality is itself observable, the underlying fault may not be as the number of internal sensors is limited.

Without loss of generality, we consider a single event of interest to diagnose, d. When there are multiple events to diagnose, the methodologies that we describe in the remaining of this entry can be applied to each event of interest d_i, $i = 1, \ldots n$, individually; in this case, the other events of interest d_j, $j \neq i$ are treated the same as the other unobservable events in the set E_{uo} in the process of model-based inferencing.

Problem Formulation

Event Diagnosis

We start with a general language-based formulation of the event diagnosis problem. The information available to the agent that monitors the system behavior and performs the task of event diagnosis is the language L_M and the set of observable events E_o, along with the specific

string $t \in P(L_M)$ that it observes at run-time. The actual string generated by the system is $s \in L_M$ where $P(s) = t$. However, as far as the monitoring agent is concerned, the actual string that has occurred could be any string in $P^{-1}(t) \cap L_M$, where P^{-1} is the inverse projection operation, i.e., $P^{-1}(t)$ is the set of all strings $s_t \in E^*$ such that $P(s_t) = t$. Let us denote this estimate set by $\mathcal{E}(t) = P^{-1}(t) \cap L_M$, where "$\mathcal{E}$" stands for "estimate." If a string $s \in L_M$ contains event d, we write that $d \in L_M$; otherwise, we write that $d \notin L_M$.

The event diagnosis problem is to synthesize a diagnostic engine that will automatically provide the following answers from the observed t and from the knowledge of L_M and E_o:

1. **Yes**, if and only if $d \in s$ for all $s \in \mathcal{E}(t)$.
2. **No**, if and only if $d \notin s$ for all $s \in \mathcal{E}(t)$.
3. **Maybe**, if and only if there exists $s_Y, s_N \in \mathcal{E}(t)$ such that $d \in s_Y$ and $d \notin s_N$.

As defined, $\mathcal{E}(t)$ is a string-based estimate. In section "Diagnosis of Automata," we discuss how to build a finite-state structure that will encode the desired answers for the above three cases when the DES M is modeled by a deterministic finite-state automaton. The resulting structure is called a *diagnoser automaton*.

Diagnosability Analysis

Diagnosability analysis consists in determining, a priori, if any and all occurrences of event d in L_M will eventually be diagnosed, in the sense that if event d occurs, then the diagnostic engine is guaranteed to eventually issue the decision "Yes." For the sake of technical simplicity, we assume hereafter that L_M is a *live* language, i.e., any trace in L_M can always be extended by one more event. In this context, we would not want the diagnostic engine to issue the decision "Maybe" for an arbitrarily long number of event occurrences after event d occurs. When this outcome is possible, we say that event d is *not diagnosable* in language L_M.

The property of *diagnosability* of DES is defined as follows. In view of the liveness assumption on language L_M, any string s_Y' that contains event d can always be extended to a longer string, meaning that it can be made "arbitrarily long"

after the occurrence of d. That is, for any s_Y' in L_M and for any $n \in \mathbb{N}$, there exists $s_Y = s_Y' t \in L_M$ where the length of t is equal to n. Event d is *not* diagnosable in language L_M if there exists such a string s_Y together with a second string s_N that does not contain event d, and such that $P(s_Y) = P(s_N)$. This means that the monitoring agent is unable to distinguish between s_Y and s_N, yet, the number of events after an occurrence of d can be made arbitrarily large in s_Y, thereby preventing diagnosis of event d within a finite number of events after its occurrence. On the other hand, if no such pair of strings (s_Y, s_N) exists in L_M, then event d is diagnosable in L_M. (The mathematically precise definition of diagnosability is available in the literature cited at the end of this entry.)

Diagnosis of Automata

We recall the definition of a deterministic finite-state automaton, or simply automaton, from article ▶ Models for Discrete Event Systems: An Overview, with the addition of a set of unobservable events as in section "Supervisory Control Under Partial Observations" in article ▶ Supervisory Control of Discrete-Event Systems. The automaton, denoted by G, is a four-tuple $G = (X, E, f, x_0)$ where X is the finite set of states, E is the finite set of events partitioned into $E = E_o \cup E_{uo}$, x_0 is the initial state, and f is the deterministic partial transition function $f : X \times E \to X$ that is immediately extended to strings $f : X \times E^* \to X$. For a DES M represented by an automaton G, L_M is the language generated by automaton G, denoted by $\mathcal{L}(G)$ and formally defined as the set of all strings for which the extended f is defined. It is an infinite set if the transition graph of G has one or more cycles. In view of the liveness assumption made on L_M in the preceding section, G has no reachable deadlocked state, i.e., for all $s \in E^*$ such that $f(x, s)$ is defined, then there exists $\sigma \in E$ such that $f(x, s\sigma)$ is also defined.

To synthesize a diagnoser automaton that correctly performs the diagnostic task formulated in the preceding section, we proceed as follows.

First, we perform the parallel composition (denoted by ||) of G with the two-state *label automaton* A_{label} that is defined as follows. $A_{label} = (\{N, Y\}, \{d\}, f_{label}, N)$, where f_{label} has two transitions defined: (i) $f_{label}(N, d) = Y$ and (ii) $f_{label}(Y, d) = Y$. The purpose of A_{label} is to record the occurrence of event d, which causes a transition to state Y. By forming $G_{labeled} = G||A_{label}$, we record in the states of $G_{labeled}$, which are of the form (x_G, x_A), if the first element of the pair, state $x_G \in X$, was reached or not by executing event d at some point in the past: if d was executed, then $x_A = Y$, otherwise $x_A = N$. (We refer the reader to Chap. 2 in Cassandras and Lafortune (2008) for the formal definition of parallel composition of automata.) By construction, $\mathcal{L}(G_{labeled}) = \mathcal{L}(G)$.

The second step of the construction of the diagnoser automaton is to build the *observer* of $G_{labeled}$, denoted by $Obs(G_{labeled})$, with respect to the set of observable events E_o. (We refer the reader to Chap. 2 in Cassandras and Lafortune (2008) for the definition of the observer automaton and for its construction.) The construction of the observer involves the standard subset construction algorithm for nondeterministic automata in automata theory; here, the unobservable events are the source of nondeterminism, since they effectively correspond to ε-transitions. The diagnoser automaton of G with respect to E_o is defined as $Diag(G) = Obs(G||A_{label})$. Its event set is E_o.

The states of $Diag(G)$ are *sets* of state pairs of the form (x_G, x_A) where x_A is either N or Y. Examination of the state of $Diag(G)$ reached by string $t \in P[\mathcal{L}(G)]$ provides the answers to the event diagnosis problem. Let us denote that state by x_{Diag}^t. Then:

1. The diagnostic decision is **Yes** if *all* state pairs in x_{Diag}^t have their second component equal to Y; we call such a state a "Yes-state" of $Diag(G)$.
2. The diagnostic decision is **No** if *all* state pairs in x_{Diag}^t have their second component equal to N; we call such a state a "No-state" of $Diag(G)$.
3. The diagnostic decision is **Maybe** if there is at least one state pair in x_{Diag}^t whose second

component is equal to Y and at least one state pair in x_{Diag}^t whose second component is equal to N; we call such a state a "Maybe-state" of $Diag(G)$.

To perform run-time diagnosis, it therefore suffices to examine the current state of $Diag(G)$. Note that $Diag(G)$ can be computed off-line from G and stored in memory, so that run-time diagnosis requires only updating the new state of $Diag(G)$ on the basis of the most recent observed event (which is necessarily in E_o). If storing the entire structure of $Diag(G)$ is impractical, its current state can be computed on-the-fly on the basis of the most recent observed event and of the transition structure of $G_{labeled}$; this involves one step of the subset construction algorithm.

As a simple example, consider the automaton G_1 shown in Fig. 1, where $E_{uo} = \{d\}$. The occurrence of event d changes the behavior of the system such that event c does not cause a return to the initial state 1 (identified by incoming arrow); instead, the system gets stuck in state 3 after d occurs.

Its diagnoser is depicted in Fig. 2. It contains one Yes-state, state $\{(3, Y)\}$ (abbreviated as "3Y" in the figure), one No-state, and two Maybe-states (similarly abbreviated). Two consecutive occurrences of event c, or an occurrence of b right after c, both indicate that the system must be in state 3,
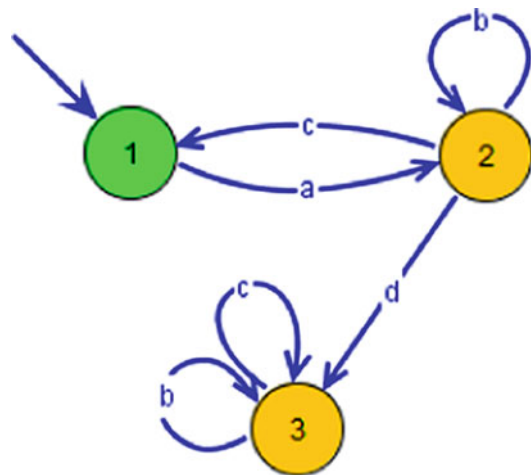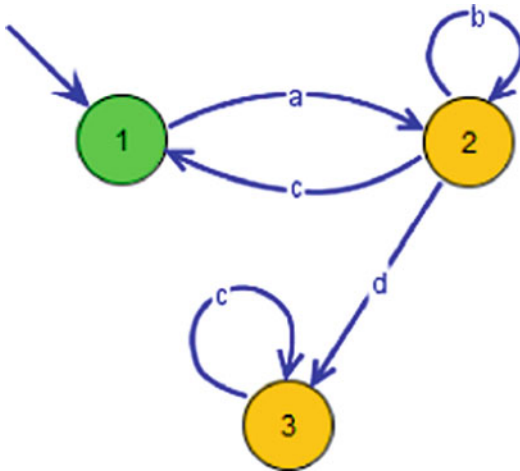

Diagnosis of Discrete Event Systems, Fig. 1
Automaton G_1
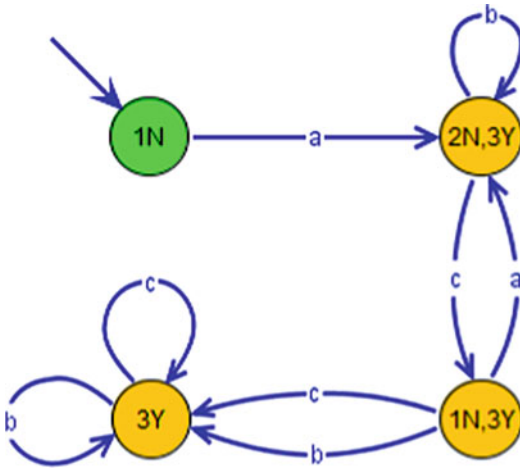
Diagnosis of Discrete Event Systems, Fig. 2
Diagnoser automaton of G_1
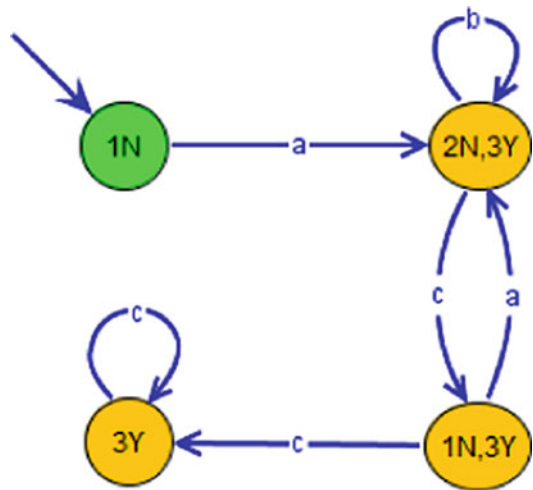
Diagnosis of Discrete Event Systems, Fig. 4
Diagnoser automaton of G_2

Diagnosis of Discrete Event Systems, Fig. 3
Automaton G_2

i.e., that event d must have occurred; this is captured by the two transitions from Maybe-state $\{(1, N), (3, Y)\}$ to the Yes-state in Diag(G_1). As a second example, consider the automaton G_2 shown in Fig. 3, where the self-loop b at state 3 in G_1 has been removed. Its diagnoser is shown in Fig. 4.

Diagnoser automata provide as much information as can be inferred, from the available observations and the automaton model of the system, regarding the past occurrence of unobservable event d. However, we may want to answer the

question: Can the occurrence of event d always be diagnosed? This is the realm of diagnosability analysis discussed in the next section.

Diagnosability Analysis of Automata

As mentioned earlier, diagnosability analysis consists in determining, a priori, if any and all occurrences of event d in L_M will eventually be diagnosed. In the case of diagnoser automata, we do not want Diag(G) to loop forever in a cycle of Maybe-states and never enter a Yes-state if event d has occurred, as happens in the diagnoser in Fig. 2 for the string adb^n when n gets arbitrarily large. In this case, Diag(G_1) loops in Maybe-state $\{(2, N), (3, Y)\}$, and the occurrence of d goes undetected. This shows that event d is not diagnosable in G_1; the counterexample is provided by strings $s_Y = adb^n$ and $s_N = ab^n$.

For systems modeled as automata, diagnosability can be tested with quadratic time complexity in the size of the state space of G by forming a so-called twin-automaton (also called "verifier") where G is parallel composed with itself, but synchronization is only enforced on *observable* events, allowing arbitrary interleavings of unobservable events. The test for diagnosability reduces to detection of cycles that occur after

event d in the twin-automaton. It can be verified that for automaton G_2 in our example, event d is diagnosable. Indeed, it is clear from the structure of G_2 that Diag(G_2) in Fig. 4 will never loop in Maybe-state $\{(2, N), (3, Y)\}$ if event d occurs; rather, after d occurs, G_2 can only execute c events, and after two such events, Diag(G_2) enters Yes-state $\{(3, Y)\}$.

Note that diagnosability will not hold in an automaton that contains a cycle of unobservable events after the occurrence of event d, although this is not the only instance where the property is violated, as we saw in our simple example.

Diagnosis and Diagnosability Analysis of Petri Nets

There are several approaches for diagnosis and diagnosability analysis of DES modeled by Petri nets, depending on the boundedness properties of the net and on what is observable about its behavior. Let N be the Petri net model of the system, which consists of a Petri net structure along with an initial marking of all places. If the transitions of N are labeled by events in a set E, some by observable events in E_o and some by unobservable events in E_{uo}, and if the contents of the Petri net places are not observed except for the initial marking of N, then we have a language-based diagnosis problem as considered so far in this entry, for language $\mathcal{L}(N)$ and for $E = E_o \cup E_{uo}$, with event of interest $d \in E_{uo}$. In this case, if the set of reachable states of the net is bounded, then we can use the reachability graph as an equivalent automaton model of the same system and build a diagnoser automaton as described earlier. It is also possible to encode diagnoser states into the original structure of net N by keeping track of all possible net markings following the observation of an event in E_o, appending the appropriate label ("N" or "Y") to each marking in the state estimate. This is reminiscent of the on-the-fly construction of the current state of the diagnoser automaton discussed earlier, except that the possible system states are directly listed as Petri net markings on the structure of N. Regarding diagnosability

analysis, it can be performed using the twin-automaton technique of the preceding section, from the reachability graph of N.

Another approach that is actively being pursued in current literature is to exploit the structure of the net model N for diagnosis and for diagnosability analysis, instead of working with the automaton model obtained from its reachability graph. In addition to potential computational gains from avoiding the explicit generation of the entire set of reachable states, this approach is motivated by the need to handle Petri nets whose sets of reachable states are infinite and in particular Petri nets that generate languages that are not regular and hence cannot be represented by finite-state automata. Moreover, in this approach, one can incorporate potential observability of token contents in the places of the Petri nets. We refer the interested reader to the relevant chapters in Campos et al. (2013) and Seatzu et al. (2013) for coverage of these topics.

Current and Future Directions

The basic methodologies described so far for diagnosis and diagnosability analysis have been extended in many different directions. We briefly discuss a few of these directions, which are active research areas. Detailed coverage of these topics is beyond the scope of this entry and is available in the references listed at the end.

Diagnosis of *timed models* of DES has been considered, for classes of timed automata and timed Petri nets, where the objective is to ensure that each occurrence of event d is detected within a bounded time delay. Diagnosis of *stochastic models* of DES has been considered, in particular stochastic automata, where the hard diagnosability constraints are relaxed and detection of each occurrence of event d must be guaranteed with some probability $1 - \epsilon$, for some small $\epsilon > 0$. Stochastic models also allow handling of unreliable sensors or noisy environments where event observations may be corrupted with some probability, such as when an occurrence of event a is observed as event a 80 % of the time and as some other event a' 20 % of the time.

Decentralized diagnosis is concerned with DES that are observed by several monitoring agents $i = 1, \ldots, n$, each with its own set of observable events $E_{o,i}$ and each having access to the entire set of system behaviors, L_M. The task is to design a set of individual diagnosers, one for each set $E_{o,i}$, such that the n diagnosers together diagnose all the occurrences of event d. In other words, for each occurrence of event d in any string of L_M, there exists at least one diagnoser that will detect it (i.e., answer "Yes"). The individual diagnosers may or may not communicate with each other at run-time or they may communicate with a coordinating diagnoser that will fuse their information; several decentralized diagnosis architectures have been studied and their properties characterized. The focus in these works is the decentralized nature of the information available about the strings in L_M, as captured by the individual observable event sets $E_{o,i}$, $i = 1, \ldots, n$.

Distributed diagnosis is closely related to decentralized diagnosis, except that it normally refers to situations where each individual diagnoser uses only part of the entire system model. Let M be an automaton G obtained by parallel composition of subsystem models: $G = \|_{i=1,n} G_i$. In distributed diagnosis, one would want to design each individual diagnoser Diag_i on the basis of G_i alone or on the basis of G_i and of an *abstraction* of the rest of the system, $\|_{j=1,n; j \neq i} G_j$. Here, the emphasis is on the distributed nature of the system, as captured by the parallel composition operation. In the case where M is a Petri net N, the distributed nature of the system may be captured by individual net models N_i, $i = 1, \ldots, n$, that are coupled by common places, i.e., *place-bordered* Petri nets.

Robust diagnosis generally refers to decentralized or distributed diagnosis, but where one or more of the individual diagnosers may fail. Thus, there must be built-in redundancy in the set of individual diagnosers so that they together may still detect every occurrence of event d even if one or more of them ceases to operate.

So far we have considered a fixed and static set of observable events, $E_o \subset E$, where every occurrence of each event in E_o is always observed by the monitoring agent. However, there are many instances where one would want the monitoring agent to dynamically activate or de-activate the observability properties of a subset of the events in E_o; this arises in situations where event monitoring is "costly" in terms of energy, bandwidth, or security reasons. This is referred to as the case of *dynamic observations*, and the goal is to synthesize sensor activation policies that minimize a given cost function while preserving the diagnosability properties of the system.

Cross-References

▶ Models for Discrete Event Systems: An Overview
▶ Modeling, Analysis, and Control with Petri Nets
▶ Supervisory Control of Discrete-Event Systems
▶ Modeling, Analysis, and Control with Petri Nets

Recommended Reading

There is a very large amount of literature on diagnosis and diagnosability analysis of DES that has been published in control engineering, computer science, and artificial intelligence journals and conference proceedings. We mention a few recent books or survey articles that are a good starting point for readers interested in learning more about this active area of research. In the DES literature, the study of fault diagnosis and the formalization of diagnosability properties started in Lin (1994) and Sampath et al. (1995). Chapter 2 of the textbook Cassandras and Lafortune (2008) contains basic results about diagnoser automata and diagnosability analysis of DES, following the approach introduced in Sampath et al. (1995). The research monograph Lamperti and Zanella (2003) presents DES diagnostic methodologies developed in the artificial intelligence literature. The survey paper Zaytoon and Lafortune (2013) presents a detailed overview of fault diagnosis research in the control engineering literature. The two edited books

Campos et al. (2013) and Seatzu et al. (2013) contain chapters specifically devoted to diagnosis of automata and Petri nets, with an emphasis on automated manufacturing applications for the latter. Specifically, Chaps. 5, 14, 15, 17, and 19 in Campos et al. (2013) and Chaps. 22–25 in Seatzu et al. (2013) are recommended for further reading on several aspects of DES diagnosis. Zaytoon and Lafortune (2013) and the cited chapters in Campos et al. (2013) and Seatzu et al. (2013) contain extensive bibliographies.

Bibliography

Campos J, Seatzu C, Xie X (eds) (2013) Formal methods in manufacturing. Series on industrial information technology. CRC/Taylor and Francis, Boca Raton, FL

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York, NY

Lamperti G, Zanella M (2003) Diagnosis of active systems: principles and techniques. Kluwer, Dordrecht

Lin F (1994) Diagnosability of discrete event systems and its applications. Discret Event Dyn Syst Theory Appl 4(2):197–212

Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D (1995) Diagnosability of discrete event systems. IEEE Trans Autom Control 40(9):1555–1575

Seatzu C, Silva M, van Schuppen J (eds) (2013) Control of discrete-event systems. Automata and Petri net perspectives. Lecture notes in control and information sciences, vol 433. Springer, London

Zaytoon J, Lafortune S (2013) Overview of fault diagnosis methods for discrete event systems. Annu Rev Control 37(2):308–320

Differential Geometric Methods in Nonlinear Control

A.J. Krener
Department of Applied Mathematics, Naval Postgrauate School, Monterey, CA, USA

Abstract

In the early 1970s, concepts from differential geometry were introduced to study nonlinear control systems. The leading researchers in this effort were Roger Brockett, Robert Hermann, Henry Hermes, Alberto Isidori, Velimir Jurdjevic, Arthur Krener, Claude Lobry, and Hector Sussmann. These concepts revolutionized our knowledge of the analytic properties of control systems, e.g., controllability, observability, minimality, and decoupling. With these concepts, a theory of nonlinear control systems emerged that generalized the linear theory. This theory of nonlinear systems is largely parallel to the linear theory, but of course it is considerably more complicated.

Keywords

Introduction

This is a brief survey of the influence of differential geometric concepts on the development of nonlinear systems theory. Section "A Primer on Differential Geometry" reviews some concepts and theorems of differential geometry. Nonlinear controllability and nonlinear observability are discussed in sections "Controllability of Nonlinear Systems" and "Observability for Nonlinear Systems". Section "Minimal Realizations" discusses minimal realizations of nonlinear systems, and section "Disturbance Decoupling" discusses the disturbance decoupling problem.

A Primer on Differential Geometry

Perhaps a better title might be "A Primer on Differential Topology" since we will not treat Riemannian or other metrics. A n-dimensional manifold $\mathcal{M}$ is a topological space that is locally homeomorphic to a subset of IR^n. For simplicity, we shall restrict our attention to smooth (C^∞) manifolds and smooth objects on them. Around each point $p \in \mathcal{M}$, there is at least one coordinate chart that is a neighborhood $\mathcal{N}_p \subset \mathcal{M}$ and a homeomorphism $x : \mathcal{N}_p \rightarrow \mathcal{U}$ where $\mathcal{U}$ is an open subset of IR^n. When two coordinate

D

charts overlap, the change of coordinates should be smooth. For simplicity, we restrict our attention to differential geometric objects described in local coordinates.

In local coordinates, a vector field is just an ODE of the form

$$\dot{x} = f(x) \tag{1}$$

where $f(x)$ is a smooth $IR^{n \times 1}$ valued function of x. In a different coordinate chart with local coordinates z, this vector field would be represented by a different formula:

$$\dot{z} = g(z)$$

If the charts overlap, then on the overlap they are related:

$$f(x(z)) = \frac{\partial x}{\partial z}(z)g(z), \ \ g(z(x)) = \frac{\partial z}{\partial x}(z)f(x)$$

Since $f(x)$ is smooth, it generates a smooth flow $\phi(t, x^0)$ where for each t, the mapping $x \mapsto \phi(t, x^0)$ is a local diffeomorphism and for each x^0 the mapping $t \mapsto \phi(t, x^0)$ is a solution of the ODE (1) satisfying the initial condition $\phi(0, x^0) = x^0$. We assume that all the flows are complete, i.e., defined for all $t \in IR$, $x \in \mathcal{M}$. The flows are one parameter groups, i.e., $\phi(t, \phi(s, x^0)) = \phi(t + s, x^0) = \phi(s, \phi(t, x^0))$.

If $f(x^0) = b$, a constant vector, then locally the flow looks like translation, $\phi(t, x^1) = x^1 + tb$. If $f(x^0) \neq 0$, then we can always choose local coordinates z so that in these coordinates the vector field is constant. Without loss of generality, we can assume that $x^0 = 0$ and that the first component of f is $f_1(0) \neq 0$. Define the local change of coordinates: $x(z) = \phi(z_1, x^1(z))$ where $x^1(z) = (0, z_2, \ldots, z_n)'$. It is not hard to that this is a local diffeomorphism and that, in z coordinates, the vector field is the first unit vector.

If $f(x^0) = 0$ let $F = \frac{\partial f}{\partial x}(x^0)$, then if all the eigenvalues of F are off the imaginary axis, then the integral curves of $f(x)$ and

$$\dot{z} = Fz \tag{2}$$

are locally topologically equivalent (Arnol'd 1983). This is the Grobman-Hartman theorem. There exists a local homeomorphism $z = h(x)$ that carries $x(t)$ trajectories into $z(s)$ trajectories in some neighborhood of $x^0 = 0$. This homeomorphism need not preserve time $t \neq s$, but it does preserve the direction of time. Whether these flows are locally diffeomorphic is a more difficult question that was explored by Poincaré. See the section on feedback linearization (Krener 2013).

If all the eigenvalues of F are in the open left half plane, then the linear dynamics (2) is globally asymptotically stable around $z^0 = 0$, i.e., if the flow of (2) is $\psi(t, z)$, then $\psi(t, z^1) \to 0$ as $t \to \infty$. Then it can be shown that the nonlinear dynamics is locally asymptotically stable, $\phi(t, x^1) \to x^0$ as $t \to \infty$ for all x^1 in some neighborhood of x^0.

One forms, $\omega(x)$, are dual to vector fields. The simplest example of a one form (also called a covector field) is the differential $dh(x)$ of a scalar-valued smooth function $h(x)$. This is the $IR^{1 \times n}$ covector field

$$\omega(x) = \left[\ \frac{\partial h}{\partial x_1}(x) \ \ldots \ \frac{\partial h}{\partial x_n}(x) \ \right]$$

Sometimes this is written as

$$\omega(x) = \sum_1^n \frac{\partial h}{\partial x_i}(x) \, dx_i$$

The most general smooth one form is of the form

$$\omega(x) = \left[\ \omega^1(x) \ \ldots \ \omega^n(x) \ \right]$$
$$= \sum_{i=1}^n \omega^i(x) \, dx_i$$

where the $\omega^i(x)$ are smooth functions. The duality between one forms and vector fields is the bilinear pairing

$$< \omega(x), f(x) > = \omega(f)(x) = \omega(x)f(x)$$
$$= \sum_{i=1}^n \omega^i(x) f_i(x)$$

Just as a vector field can be thought of as a first-order ODE, a one form can be thought of as a first-order PDE. Given $\omega(x)$, find $h(x)$ such that $dh(x) = \omega(x)$. A one form $\omega(x)$ is said to be exact if there exists such an $h(x)$. Of course if there is one solution, then there are many all differing by a constant of integration which we can take as the value of h at some point x^0.

Unlike smooth first-order ODEs, smooth first-order PDEs do not always have a solution. There are integrability conditions and topological conditions that must be satisfied. Suppose $dh(x) = \omega(x)$, then $\frac{\partial h}{\partial x_i}(x) = \omega^i(x)$ so

$$\frac{\partial \omega^i}{\partial x_j}(x) = \frac{\partial^2 h}{\partial x_j \partial x_i}(x) = \frac{\partial^2 h}{\partial x_i \partial x_j}(x) = \frac{\partial \omega^j}{\partial x_i}(x)$$

Therefore, for the PDE to have a solution, the integrability conditions

$$\frac{\partial \omega^i}{\partial x_j}(x) - \frac{\partial \omega^j}{\partial x_i}(x) = 0$$

must be satisfied. The exterior derivative of a one form is a skew-symmetric matrix field

$$d\omega(x) = \sum_{i<j} \left(\frac{\partial \omega^i}{\partial x_j}(x) - \frac{\partial \omega^j}{\partial x_i}(x) \right) dx_i \wedge dx_j$$

A one form $\omega(x)$ is said to be closed if $d\omega(x) = 0$. This is locally sufficient for there to exist an $h(x)$ such that $dh(x) = \omega(x)$.

Every exact form is closed but not every closed form is exact. A counter example on IR^2 is

$$\omega(x) = \begin{bmatrix} -x_2 & x_1 \end{bmatrix}$$

This is closed but not exact. The line integral of this convector field around any circle centered at the origin is 2π. If it were exact, the line integral would have been zero because the curve ends where it begins.

The Lie derivative of a scalar-valued function $h(x)$ by a vector field $f(x)$ is denied to be the scalar-valued function

$$L_f(h)(x) = \frac{\partial h}{\partial x}(x) f(x) = \; <dh(x), f(x)>$$

This can be iterated

$$L_f^k(h)(x) = \frac{\partial L_f^{k-1}h}{\partial x}(x) f(x)$$

If $h(x) \in IR^{p\times 1}$, then $L_f(h)(x) \in IR^{p\times 1}$.

The Lie bracket of two vector fields $f^1(x)$ and $f^2(x)$ is another vector field

$$\left[f^1, f^2\right](x) = \frac{\partial f^2}{\partial x}(x) f^1(x) - \frac{\partial f^1}{\partial x}(x) f^2(x)$$

Clearly, the Lie bracket is skew symmetric, $\left[f^1, f^2\right](x) = -\left[f^2, f^1\right](x)(x)$. It also satisfies the Jacobi identity

$$\left[f^1, \left[f^2, f^3\right]\right](x) + \left[f^2, \left[f^3, f^1\right]\right](x)$$
$$+ \left[f^3, \left[f^1, f^2\right]\right](x) = 0$$

Repeated Lie brackets are often expressed inductively as

$$ad_f^0(g)(x) = g(x)$$
$$ad_f^k(g)(x) = \left[f, ad_f^{k-1}(g)\right](x)$$

The geometric interpretation of the Lie bracket $[f, g](x)$ is the infinitesimal commutator of their flows $\phi(t, x)$ and $\psi(t, x)$, i.e.,

$$\psi(t, \phi(t, x)) - \phi(t, \psi(t, x)) = [f, g](x) t^2$$
$$+ O(t)^3$$

Another interpretation of the Lie bracket is given by the Lie series expansion

$$g(\phi(t, x)) = \sum_{k=0}^{\infty} (-1)^k ad_f^k(g)(x) \frac{t^k}{k!}$$

This is a Taylor series expansion which is convergent for small $|t|$ if f, g are real analytic vector fields. Another Lie series is

$$h(\phi(t, x)) = \sum_{k=0}^{\infty} L_f^k(h)(x) \frac{t^k}{k!}$$

Given a smooth mapping $z = \theta(x)$ from an open subset of IR^n to an open subset of IR^m and vector fields $f(x)$ and $g(z)$ on these open subsets, we sat $f(x)$ is θ-related to $g(z)$ if

$$g(\theta(x)) = \frac{\partial \theta}{\partial x}(x) f(x)$$

It is not hard to see that if $f^1(x)$, $f^2(x)$ are θ-related to $g^1(z)$, $g^2(z)$, then $\left[f^1, f^2\right](x)$ is θ-related to $\left[g^1, g^2\right](z)$. For this reason, we say that the Lie bracket is an intrinsic differentiation. The other intrinsic differentiation is the exterior derivative operation d.

The Lie derivative of a one form $\omega(x)$ by a vector field $f(x)$ is given by

$$L_f(\omega)(x) = \sum_{i,j} \left(\frac{\partial \omega^i}{\partial x_j}(x) f_j(x) \right.$$
$$\left. + \omega_j(x) \frac{\partial f_j}{\partial x_i}(x) \right) \, dx_i$$

It is not hard to see that

$$L_f(<\omega, g>)(x) = <L_f(\omega), g>(x)$$
$$+ <\omega, [f, g]>(x)$$

and

$$L_f(dh)(x) = d(L_f(h))(x) \qquad (3)$$

Control systems involve multiple vector fields. A distribution $\mathcal{D}$ is a set of vector fields on $\mathcal{M}$ that is closed under addition of vector fields and under multiplication by scalar functions. A distribution defines at each $x \in \mathcal{M}$ a subspace of the tangent space

$$D(x) = \{f(x) : f \in \mathcal{D}\}$$

These subspaces form a subbundle D of the tangent bundle. If the subspaces are all of the same dimension, then the distribution is said to be nonsingular. We will restrict our attention to nonsingular distributions.

A codistribution (or Pfaffian system) $\mathcal{E}$ is a set of one forms on $\mathcal{M}$ that is closed under addition and multiplication by scalar functions.

A codistribution defines at each $x \in \mathcal{M}$ a subspace of the cotangent space

$$\{\omega(x) : \omega \in \mathcal{E}\}$$

These subspaces form a subbundle E of the cotangent bundle. If the subspaces are all of the same dimension, then the codistribution is said to be nonsingular. Again, we will restrict our attention to nonsingular codistributions.

Every distribution $\mathcal{D}$ defines a dual codistribution

$$\mathcal{D}^* = \{\omega(x) : \omega(x) f(x) = 0, \text{ for all } f(x) \in \mathcal{D}\}$$

and vice versa

$$\mathcal{E}_* = \{f(x) : \omega(x) f(x) = 0, \text{ for all } \omega(x) \in \mathcal{E}\}$$

A k dimensional distribution $\mathcal{D}$ (or its dual codistribution $\mathcal{D}^*$) can be thought of as a system of PDEs on $\mathcal{M}$. Find $n-k$ independent functions $h_1(x), \ldots, h_{n-k}(x)$ such that

$$dh_i(x) f(x) = 0 \text{ for all } f(x) \in \mathcal{D}$$

The functions $h_1(x), \ldots, h_{n-k}(x)$ are said to be independent if $dh_1(x), \ldots, dh_{n-k}(x)$ are linearly independent at every $x \in \mathcal{M}$. In other words, $dh_1(x), \ldots, dh_{n-k}(x)$ span $\mathcal{D}^*$ over the space of smooth functions.

The Frobenius theorem gives the integrability conditions for these functions to exist locally. The distribution $\mathcal{D}$ must be involutive, i.e., closed under the Lie bracket,

$$[\mathcal{D}, \mathcal{D}] = \{[f, g] : f, g \in \mathcal{D}\} \subset \mathcal{D}$$

When the functions exist, their joint level sets $\{x : h_i(x) = c_i, i = 1, \ldots, n - k\}$ are the leaves of a local foliation. Through each x^0 in a convex local coordinate chart $\mathcal{N}$, there exists locally a a k-dimensional submanifold $\{x \in \mathcal{N} : h_i(x) = h_i(x^0)\}$. At each x^1 in this submanifold, its tangent space is $D(x)$. Whether these $h_i(x)$ exist globally to define a global foliation, a partition of $\mathcal{M}$ into smooth submanifolds, is a delicate question. Consider a distribution on IR^2 generated by a constant vector field $f(x) = b$

of irrational slope, b_2/b_1 is irrational. Construct the torus T^2 as the quotient of IR^2 by the integer lattice Z^2. The distribution passes to the quotient and since it is one dimensional, it is clearly involutive. The leaves of the quotient distribution are curves that wind around the torus indefinitely, and each curve is dense in T^2. Therefore, any smooth function $h(x)$ that is constant on such a leaf is constant on all of T^2. Hence, the local foliation does not extend to a global foliation.

Another delicate question is whether the quotient space of $\mathcal{M}$ by a foliation induced by an involutive distribution is a smooth manifold (Sussmann 1975). This is always true locally, but it may not hold globally. Think of the foliation of T^2 discussed above.

Given $k \leq n$ vector fields $f^1(x), \ldots, f^k(x)$ that are linearly independent at each x and that commute, $\left[f^i, f^j\right](x) = 0$, there exists a local change of coordinates $z = z(x)$ so that in the new coordinates the vector fields are the first k unit vectors.

The involutive closure $\bar{\mathcal{D}}$ of $\mathcal{D}$ is the smallest involutive distribution containing $\mathcal{D}$. As with all distributions, we always assume implicitly that is nonsingular. A point x^1 is $\mathcal{D}$-accessible from x^0 if there exists a continuous and piecewise smooth curve joining x^0 to x^1 whose left and right tangent vectors are always in D. Obviously $\mathcal{D}$-accessibility is an equivalence relation. Chow's theorem (1939) asserts that its equivalence classes are the leaves of the foliation induced by $\bar{\mathcal{D}}$. Chow's theorem goes a step further. Suppose $f^1(x), \ldots, f^k(x)$ span $D(x)$ at each $x \in \mathcal{M}$ then given any two points x^0, x^1 in a leaf of $\bar{\mathcal{D}}$, there is a continuous and piecewise smooth curve joining x^0 to x^1 whose left and right tangent vectors are always one of the $f^i(x)$.

Controllability of Nonlinear Systems

An initialized nonlinear system that is affine in the control is of the form

$$
\begin{aligned}
\dot{x} &= f(x) + g(x)u \\
&= f(x) + \sum_{j=1}^{m} g^j(x)u_j \\
y &= h(x) \\
x(0) &= x^0
\end{aligned}
\tag{4}
$$

where the state x are local coordinates on an n-dimensional manifold $\mathcal{M}$, the control u is restricted to lie in some set $\mathcal{U} \subset IR^m$, and the output y takes values IR^p. We shall only consider such systems.

A particular case is a linear system of the form

$$
\begin{aligned}
\dot{x} &= Fx + Gu \\
y &= Hx \\
x(0) &= x^0
\end{aligned}
\tag{5}
$$

where $\mathcal{M} = IR^n$ and $\mathcal{U} = IR^m$.

The set $\mathcal{A}_t(x^0)$ of points accessible at time $t \geq 0$ from x^0 is the set of all $x^1 \in \mathcal{M}$ such that there exists a bounded, measurable control trajectory $u(s) \in \mathcal{U}$, $0 \leq s \leq t$, so that the solution of (4) satisfies $x(t) = x^1$. We define $\mathcal{A}(x^0)$ as the union of $\mathcal{A}_t(x^0)$ for all $t \geq 0$. The system (4) is said to be controllable at time $t > 0$ if $\mathcal{A}_t(x^0) = \mathcal{M}$ and controllable in forward time if $\mathcal{A}(x^0) = \mathcal{M}$.

For linear systems, controllability is a rather straightforward matter, but for nonlinear systems it is more subtle with numerous variations. The variation of constants formula gives the solution of the linear system (5) as

$$
x(t) = e^{Ft}x^0 + \int_0^t e^{F(t-s)}Gu(s)\,ds
$$

so $\mathcal{A}_t(x^0)$ is an affine subspace of IR^n for any $t > 0$. It is not hard to see that the columns of

$$
\left[\, G \ldots F^{n-1}G \,\right]
\tag{6}
$$

are tangent to this affine subspace, so if this matrix is of rank n, then $\mathcal{A}_t(x^0) = IR^n$ for any $t > 0$. This is the so-called controllability rank condition for linear systems.

Turning to the nonlinear system (4), let $\mathcal{D}$ be the distribution spanned by the vector fields $f(x), g^1(x), \ldots, g^m(x)$, and let $\bar{\mathcal{D}}$ be its involutive closure. It is clear that $\mathcal{A}(x^0)$ is contained in the leaf of $\bar{\mathcal{D}}$ through x^0. Krener (1971) showed that $\mathcal{A}(x^0)$ has nonempty interior in this leaf. More precisely, $\mathcal{A}(x^0)$ is between an open set and its closure in the relative topology of the leaf.

D

Let $\mathcal{D}_0$ be the smallest distribution containing the vector fields $g^1(x), \ldots, g^m(x)$ and invariant under bracketing by $f(x)$, i.e.,

$$[f, \mathcal{D}_0] \subset \mathcal{D}_0$$

and let $\bar{\mathcal{D}}_0$ be its involutive closure. Sussmann and Jurdjevic (1972) showed that $\mathcal{A}_t(x^0)$ is in the leaf of $\bar{\mathcal{D}}_0$ through x^0, and it is between an open set and its closure in the topology of this leaf.

For linear systems (5) where $f(x) = Fx$ and $g^j(x) = G^j$, the jth column of G, it is not hard to see that

$$ad_f^k(g^j)(x) = (-1)^k F^k G^j$$

so the nonlinear generalization of the controllability rank condition is that at each x the dimension of $\bar{D}_0(x)$ is n. This guarantees that $\mathcal{A}_t(x^0)$ is between an open set and its closure in the topology of $\mathcal{M}$.

The condition that

$$\text{dimension } \bar{D}(x) = n \tag{7}$$

is referred to as the nonlinear controllability rank condition. This guarantees that $\mathcal{A}(x^0)$ is between an open set and its closure in the topology of $\mathcal{M}$.

There are stronger interpretations of controllability for nonlinear systems. One is short time local controllability (STLC). The definition of this is that the set of accessible points from x^0 in any small $t > 0$ with state trajectories restricted to an arbitrarily small neighborhood of x^0 should contain x^0 in its interior. Hermes (1994) and others have done work on this.

Observability for Nonlinear Systems

Two possible initial states x^0, x^1 for the nonlinear system are distinguishable if there exists a control $u(\cdot)$ such that the corresponding outputs $y^0(t)$, $y^1(t)$ are not equal. They are short time distinguishable if there is an $u(\cdot)$ such that $y^0(t) \neq y^1(t)$ for all small $t > 0$. They are locally short time distinguishable if in addition the corresponding state trajectories do not leave an

arbitrarily small open set containing x^0, x^1. The open set need not be connected. A nonlinear system is (short time, locally short time) observable if every pair of initial states is (short time, locally short time) distinguishable. Finally, a nonlinear system is (short time, locally short time) locally observable if every x^0 has a neighborhood such that every other point x^1 in the neighborhood is (short time, locally short time) distinguishable from x^0.

For a linear system (5), all these definitions coalesce into a single concept of observability which can be checked by the observability rank condition which is that the rank of

$$\begin{bmatrix} H \\ HF \\ \vdots \\ HF^{n-1} \end{bmatrix} \tag{8}$$

equals n.

The corresponding concept for nonlinear systems involves $\mathcal{E}$, the smallest codistribution containing $dh_1(x), \ldots, dh_p(x)$ that is invariant under repeated Lie differentiation by the vector fields $f(x), g^1(x), \ldots, g^m(x)$. Let

$$E(x) = \{\omega(x) : \omega \in \mathcal{E}\}$$

The nonlinear observability rank condition is

$$\text{dimension } E(x) = n \tag{9}$$

for all $x \in \mathcal{M}$. This condition guarantees that the nonlinear system is locally short time, locally observable. It follows from (3) that $\mathcal{E}$ is spanned by a set of exact one form, so its dual distribution $\mathcal{E}^*$ is involutive.

For a linear system (5), the input–output mapping from $x(0) = x^i, i = 0, 1$ is

$$y^i(t) = He^{Ft}x^i + \int_0^t He^{F(t-s)}Gu(s)\, ds$$

The difference is

$$y1(t) - y^0(t) = He^{Ft}(x^1 - x^0)$$

So if one input $u(\cdot)$ distinguishes x^0 from x^1, then so does every input.

For nonlinear systems, this is not necessarily true. That prompted Gauthier et al. (1992) to introduce a stronger concept of observability for nonlinear systems. For simplicity, we describe it for scalar input and scalar output systems. A nonlinear system is uniformly observable for any input if there exist local coordinates so that it is of the form

$$y = x_1 + h(u)$$
$$\dot{x}_1 = x_2 + f_1(x_1, u)$$
$$\vdots$$
$$\dot{x}_{n-1} = x_n + f_{n-1}(x_1, \ldots, x_{n-2}, u)$$
$$\dot{x}_n = f_n(x, u)$$

Cleary if we know $u(\cdot)$, $y(\cdot)$, then by repeated differentiation of $y(\cdot)$, we can reconstruct $x(\cdot)$. It has been shown that for nonlinear systems that are uniformly observable for any input, the extended Kalman filter is a locally convergent observer (Krener 2002a), and the minimum energy estimator is globally convergent (Krener 2002b).

Minimal Realizations

The initialized nonlinear system (4) can be viewed as defining a input–output mapping from input trajectories $u(\cdot)$ to output trajectories $y(\cdot)$. Is it a minimal realization of this mapping, does there exists an initialized nonlinear system on a smaller dimensional state space that realizes the same input–output mapping?

Kalman showed that (5) initialized at $x^0 = 0$ is minimal iff the controllability rank condition and the observability rank condition hold. He also showed how to reduce a linear system to a minimal one.

If the controllability rank condition does not hold, then the span of (6) dimension is $k < n$. This subspace contains the columns of G and is invariant under multiplication by F. In fact, it is the maximal subspace with these properties. So the linear system can be restricted to

this k-dimensional subspace, and it realizes the same input–output mapping from $x^0 = 0$. The restricted system satisfies the controllability rank condition.

If the observability rank condition does not hold, then the kernel of (8) is a subspace of $\mathit{IR}^{n\times 1}$ of dimension $n - l > 0$. This subspace is in the kernel of H and is invariant under multiplication by F. In fact, it is the maximal subspace with these properties. Therefore, there is a quotient linear system on the $\mathit{IR}^{n\times 1}$ mod, the kernel of (8) which has the same input–output mapping. The quotient is of dimension $l < n$, and it realizes the same input–output mapping. The quotient system satisfies the observability rank condition.

By employing these two steps in either order, we pass to a minimal realization of the input–output map of (5) from $x^0 = 0$. Kalman also showed that two linear minimal realizations differ by a linear change of state coordinates.

An initialized nonlinear system is a realization of minimal dimension of its input–output mapping if the nonlinear controllability rank condition (7) and the nonlinear observability rank condition (9) hold.

If the nonlinear controllability rank condition (7) fails to hold because the dimension of $D(x)$ is $k < n$, then by replacing the state space $\mathcal{M}$ with the k dimensional leaf through x^0 of the foliation induced by $\mathcal{D}$, we obtain a smaller state space on which the nonlinear controllability rank condition (7) holds. The input–output mapping is unchanged by this restriction.

Suppose the nonlinear observability rank condition (9) fails to hold because the dimension of $E(x)$ is $l < n$. Then consider a convex neighborhood $\mathcal{N}$ of x^0. The distribution $\mathcal{E}^*$ induces a local foliation of $\mathcal{N}$ into leaves of dimension $n - l > 0$. The nonlinear system leaves this local foliation invariant in the following sense. Suppose x^0 and x^1 are on the same leaf then if $x^i(t)$ is the trajectory starting at x^i, then $x^0(t)$ and $x^1(t)$ are on the same leaf as long as the trajectories remain in $\mathcal{N}$. Furthermore, $h(x)$ is constant on leaves so $y^0(t) = y^1(t)$. Hence, there exists locally a nonlinear system whose state space is the leaf space. On this leaf space, the nonlinear observability rank condition holds (9),

and the projected system has the same input–output map as the original locally around x^0. If the leaf space of the foliation induced by $\mathcal{E}^*$ admits the structure of a manifold, then the reduced system can be defined globally on it. Sussmann (1973) and Sussmann (1977) studied minimal realizations of analytic nonlinear systems.

The state space of two minimal nonlinear systems need not be diffeomorphic. Consider the system

$$\dot{x} = u, \;\; y = \sin x$$

where x, u are scalars. We can take the state space to be either $\mathcal{M} = I\!R$ or $\mathcal{M} = S^1$, and we will realize the same input–output mapping. These two state spaces are certainly not diffeomorphic but one is a covering space of the other.

Lie Jet and Approximations

Consider two initialized nonlinear controlled dynamics

$$\dot{x} = f^0(x) + \sum_{j=1}^m f^j(x)u_j \quad (10)$$
$$x(0) = x^0$$

$$\dot{z} = g^0(z) + \sum_{j=1}^m g^j(z)u_j \quad (11)$$
$$z(0) = z^0$$

Suppose that (10) satisfies the nonlinear controllability rank condition (7). Further, suppose that there is a smooth mapping $\Phi(x) = z$ and constants $M > 0, \epsilon > 0$ such that for any $\|u(t)\| < 1$, the corresponding trajectories $x(t)$ and $z(t)$ satisfy

$$\|\Phi(x(t)) - z(t)\| < Mt^{k+1} \quad (12)$$

for $0 \le t < \epsilon$.

Then it is not hard to show that the linear map $L = \frac{\partial \Phi}{\partial x}(x^0)$ takes brackets up to order k of the vector fields f^j evaluated at x^0 into the corresponding brackets of the vector fields g^j evaluated at z^0,

$$L[f^{j_l}[\ldots[f^{j_2}, f^{j_1}]\ldots]](x^0)$$
$$= [g^{j_l}[\ldots[g^{j_2}, g^{j_1}]\ldots]](z^0) \quad (13)$$

for $1 \le l \le k$.

On the other hand, if there is a linear map L such that (13) holds for $1 \le l \le k$, then there exists a smooth mapping $\Phi(x) = z$ and constants $M > 0, \epsilon > 0$ such that for any $\|u(t)\| < 1$, the corresponding trajectories $x(t)$ and $z(t)$ satisfy (12).

The k-Lie jet of (10) at x^0 is the tree of brackets $[f^{j_l}[\ldots[f^{j_2}, f^{j_1}]\ldots]](x^0)$ for $1 \le l \le k$. In some sense these are the coordinate-free Taylor series coefficients of (10) at x^0.

The dynamics (10) is free-nilpotent of degree k if all these brackets are as linearly independent as possible consistent with skew symmetry and the Jacobi identity and all higher degree brackets are zero. If it is free to degree k, the controlled dynamics (10) can be used to approximate any other controlled dynamics (11) to degree k. Because it is nilpotent, integrating (10) reduces to repeated quadratures. If all brackets with two or more f^i, $1 \le i \le m$ are zero at x^0, then (10) is linear in appropriate coordinates. If all brackets with three or more f^i, $1 \le i \le m$ are zero at x^0, then (10) is quadratic in appropriate coordinates. The references Krener and Schaettler (1988) and Krener (2010a,b) discuss the structure of the reachable sets for such systems.

Disturbance Decoupling

Consider a control system affected by a disturbance input $w(t)$

$$\dot{x} = f(x) + g(x)u + b(x)w$$
$$y = h(x) \quad (14)$$

The disturbance decoupling problem is to find a feedback $u = \kappa(x)$, so that in the closed-loop system, the output $y(t)$ is not affected by the disturbance $w(t)$. Wonham and Morse (1970) solved this problem for a linear system

$$\dot{x} = Fx + Gu + Bw$$
$$y = Hx \quad (15)$$

To do so, they introduced the concept of an F, G invariant subspace. A subspace $\mathcal{V} \subset IR^n$ is F, G invariant if

$$F\mathcal{V} \subset \mathcal{V} + \mathcal{G} \qquad (16)$$

where $\mathcal{G}$ is the span of the columns of G. It is easy to see that $\mathcal{V}$ is F, G invariant iff there exists a $K \in IR^{m \times n}$ such that

$$(F + GK)\mathcal{V} \subset \mathcal{V} \qquad (17)$$

The feedback gain K is called a friend of $\mathcal{V}$.

It is easy from (16) that if $\mathcal{V}^i$ is F, G invariant for $i = 1, 2$, then $\mathcal{V}^1 + \mathcal{V}^2$ is also. So there exists a maximal F, G invariant subspace $\mathcal{V}^{\max}$ in the kernel of H. Wonham and Morse showed that the linear disturbance decoupling problem is solvable iff $\mathcal{B} \subset \mathcal{V}^{\max}$ where $\mathcal{B}$ is the span of the columns of B.

Isidori et al. (1981a) and independently Hirschorn (1981) solved the nonlinear disturbance decoupling problem. A distribution $\mathcal{D}$ is locally f, g invariant if

$$\begin{aligned} [f, \mathcal{D}] &\subset \mathcal{D} + \Gamma \\ [g^j, \mathcal{D}] &\subset \mathcal{D} + \Gamma \end{aligned} \qquad (18)$$

for $j = 1, \ldots, m$ where Γ is the distribution spanned by the columns of g. In Isidori et al. (1981b) it is shown that if $\mathcal{D}$ is locally f, g invariant, then so is its involutive closure.

A distribution $\mathcal{D}$ is f, g invariant if there exists $\alpha(x) \in IR^{m \times 1}$ and invertible $\beta(x) \in IR^{m \times m}$ such that

$$\begin{aligned} [f + g\alpha, \mathcal{D}] &\subset \mathcal{D} \\ \left[\sum_j g^j \beta_j^k, \mathcal{D}\right] &\subset \mathcal{D} \end{aligned} \qquad (19)$$

for $k = 1, \ldots, m$. It is not hard to see that a f, g invariant is locally f, g invariant. It is shown in Isidori et al. (1981b) that if $\mathcal{D}$ is a locally f, g invariant distribution, then locally there exists $\alpha(x)$ and $\beta(x)$ so that (19) holds. Furthermore, if the state space is simply connected, then $\alpha(x)$ and $\beta(x)$ exist globally, but the matrix field $\beta(x)$ may fail to be invertible at some x.

From (18), it is clear that if $\mathcal{D}^i$ is locally f, g invariant for $i = 1, 2$, then so is $\mathcal{D}^1 + \mathcal{D}^2$. Hence, there exists a maximal locally f, g invariant

distribution $\mathcal{D}^{\max}$ in the kernel of dh. Moreover, this distribution is involutive. The disturbance decoupling problem is locally solvable iff columns of $b(x)$ are contained in $\mathcal{D}^{\max}$. If $\mathcal{M}$ is simply connected, then the disturbance decoupling problem is globally solvable iff columns of $b(x)$ are contained in $\mathcal{D}^{\max}$.

Conclusion

We have briefly described the role that differential geometric concepts played in the development of controllability, observability, minimality, approximation, and decoupling of nonlinear systems.

Cross-References

▶ Feedback Linearization of Nonlinear Systems
▶ Lie Algebraic Methods in Nonlinear Control
▶ Nonlinear Zero Dynamics

Bibliography

Arnol'd VI (1983) Geometrical methods in the theory of ordinary differential equations. Springer, Berlin

Brockett RW (1972) Systems theory on group manifods and coset spaces. SIAM J Control 10: 265–284

Chow WL (1939) Uber Systeme von Linearen Partiellen Differentialgleichungen Erster Ordnung. Math Ann 117:98–105

Gauthier JP, Hammouri H, Othman S (1992) A simple observer for nonlinear systems with applications to bioreactors. IEEE Trans Autom Control 37: 875–880

Griffith EW, Kumar KSP (1971) On the observability of nonlinear systems, I. J Math Anal Appl 35: 135–147

Haynes GW, Hermes H (1970) Non-linear controllability via Lie theory SIAM J Control 8: 450–460

Hermann R, Krener AJ (1977) Nonlinear controllability and observability. IEEE Trans Autom Control 22:728–740

Hermes H (1994) Large and small time local controllability. In: Proceedings of the 33rd IEEE conference on decision and control, vol 2, pp 1280–1281

Hirschorn RM (1981) (A,B)-invariant distributions and the disturbance decoupling of nonlinear systems. SIAM J Control Optim 19:1–19

Isidori A, Krener AJ, Gori Giorgi C, Monaco S (1981a) Nonlinear decoupling via feedback: a differential

geometric approach. IEEE Trans Autom Control 26:331–345

Isidori A, Krener AJ, Gori Giorgi C, Monaco S (1981b) Locally (f,g) invariant distributions. Syst Control Lett 1:12–15

Kostyukovskii YML (1968a) Observability of nonlinear controlled systems. Autom Remote Control 9:1384–1396

Kostyukovskii YML (1968b) Simple conditions for observability of nonlinear controlled systems. Autom Remote Control 10:1575-1584-1396

Kou SR, Elliot DL, Tarn TJ (1973) Observability of nonlinear systems. Inf Control 22: 89–99

Krener AJ (1971) A generalization of the Pontryagin maximal principle and the bang-bang principle. PhD dissertation, University of California, Berkeley

Krener AJ (1974) A generalization of Chow's theorem and the bang-bang theorem to nonlinear control problems. SIAM J Control 12:43–52

Krener AJ (1975) Local approximation of control systems. J Differ Equ 19:125–133

Krener AJ (2002a) The convergence of the extended Kalman filter. In: Rantzer A, Byrnes CI (eds) Directions in mathematical systems theory and optimization. Springer, Berlin, pp 173–182. Corrected version available at arXiv:math.OC/0212255 v. 1

Krener AJ (2002b) The convergence of the minimum energy estimator. I. In: Kang W, Xiao M, Borges C (eds) New trends in nonlinear dynamics and control, and their applications. Springer, Heidelberg, pp 187–208

Krener AJ (2010a) The accessible sets of linear free nilpotent control systems. In: Proceeding of NOLCOS 2010, Bologna

Krener AJ (2010b) The accessible sets of quadratic free nilpotent control systems. Commun Inf Syst 11:35–46

Krener AJ (2013) Feedback linearization of nonlinear systems. Baillieul J, Samad T (eds) Encyclopedia of systems and control. Springer

Krener AJ, Schaettler H (1988) The structure of small time reachable sets in low dimensions. SIAM J Control Optim 27:120–147

Lobry C (1970) Cotrollabilite des Systemes Non Lineaires. SIAM J Control 8:573–605

Sussmann HJ (1973) Minimal realizations of nonlinear systems. In: Mayne DQ, Brockett RW (eds) Geometric methods in systems theory. D. Ridel, Dordrecht

Sussmann HJ (1975) A generalization of the closed subgroup theorem to quotients of arbitrary manifolds. J Differ Geom 10:151–166

Sussmann HJ (1977) Existence and uniqueness of minimal realizations of nonlinear systems. Math Syst Theory 10:263–284

Sussmann HJ, Jurdjevic VJ (1972) Controllability of nonlinear systems. J Differ Equ 12:95–116

Wonham WM, Morse AS (1970) Decoupling an pole assignment in linear multivariable systems: a geometric approach. SIAM J Control 8:1–18

Disaster Response Robot

Satoshi Tadokoro
Tohoku University, Sendai, Japan

Abstract

Disaster response robots are robotic systems used for preventing the worsening of disaster damage under emergent situations. Robots for natural disasters (water disaster, volcano eruption, earthquakes, landslides, and fire) and man-made disasters (explosive ordnance disposal, CBRNE disasters, Fukushima Daiichi nuclear power plant accident) are introduced. Technical challenges are described on the basis of generalized data flow.

Keywords

Rescue robot; Response robot

Introduction

Disaster response robots are robotic systems used for preventing the worsening of disaster damage under emergent situations, such as for search and rescue, recovery construction, etc.

A disaster changes its state as time passes. The state starts as an unforeseen occurrence and proceeds to prevention phase, emergency response phase, recovery phase, and revival phase. Although a disaster response robot usually means a system for disaster response and recovery in a narrow sense a system used in every phase of disaster can be called a disaster response robot in a broad sense.

When parties of firefighters and military personnel respond to disasters, robots are among the technical equipments used. The purposes of robots are (1) to perform tasks that are impossible/difficult to perform by humans and conventional equipment, (2) to reduce responders' risk of inflicting secondary damage, and (3) to improve rapidity/efficiency of tasks, by using remote/automatic robot equipment.

Response Robots for Natural Disasters

Water Disaster

Underwater robots (ROV, remotely operated vehicle) are deployed to responder organizations in preparation for water damage such as caused by tsunami, flood, cataract, and accidents in the sea and rivers. They are equipped with cameras and sonars and remotely controlled by crews via tether from land or shipboard within several tens of meters area for victim search and damage investigation. After the Great Eastern Japan Earthquake in 2011, Self Defense Force and volunteers of International Rescue System Institute (IRS) and Center for Robot-Assisted Search and Rescue (CRASAR) used various types of ROVs such as SARbot shown in Fig. 1 for victim search and debris investigation in the port.

Volcano Eruption

In order to reduce risk in monitoring and recovery construction at volcano eruptions, application of robotics and remote systems is highly desired. Various types of UAVs (unmanned aerial vehicles) such as small-sized robot helicopters and airplanes have been used for this purpose.

An unmanned construction system consists of teleoperated robot backhoes, trucks, and bulldozers with wireless relaying cars and camera vehicles as shown in Fig. 2 and is remotely controlled from an operator vehicle. It has been used since the 1990s for remote civil engineering works from a distance of a few kilometers.

Structural Collapse by Earthquakes, Landslides, etc.

Small-sized UGVs (unmanned ground vehicles) were developed for victim search and monitoring in confined spaces of collapsed buildings and underground structures. VGTV X-treme shown in Fig. 3 is a tracked vehicle remotely operated via a tether. It was used for victim search at mine accidents and the 9/11 terror attack. Active scope camera shown in Fig. 4 is a serpentine robot like a fiberscope and has been used for forensic investigation of structural collapse accidents.

Fire

Large-scale fires in chemical plants and forests sometimes have a high risk, and firefighters cannot approach near them. Remote-controlled robots with firefighting nozzles for water and chemical extinguishing agents are deployed.

Disaster Response Robot, Fig. 1 SARbot (Courtesy of SeaBotix Inc.) http://www.seabotix.com/products/sarbot.htm

Disaster Response Robot, Fig. 2 Unmanned construction system (Courtesy of Society for Unmanned Construction Systems) http://www.kenmukyou.gr.jp/f_souti.htm

Disaster Response Robot, Fig. 3 VGTV X-treme (Courtesy of Recce Robotics) http://www.recce-robotics.com/vgtv.html

Disaster Response Robot, Fig. 4 Active scope camera (Courtesy of International Rescue System Institute)

Large-sized robots can discharge large volumes of the fluid with water cannons, whereas small-sized robots have better mobility.

Response Robots for Man-Made Disasters

Explosive Ordnance Disposal (EOD)

Detection and disposal of explosive ordnance is one of the most dangerous tasks. TALON, PackBot, and Telemax are widely used in military and explosive ordnance disposal teams worldwide. Telemax has an arm with seven degrees of freedom on a tracked vehicle with four sub-tracks as shown in Fig. 5. It can observe narrow spaces like overhead lockers of airplanes and bottom of automobiles by cameras, manipulate objects by the arm, and deactivate explosives by a disrupter.

CBRNE Disasters

CBRNE (chemical, biological, radiological, nuclear, and explosive) disasters have a high risk and can cause large-scale damage because human cannot detect contamination by the Hazmat (hazardous materials). Application of robotic systems is highly expected for this disaster. PackBot has sensors for toxic industrial chemicals (TIC), blood agents, blister agents, volatile organic compounds (VOCs), radiation, etc., as options and can measure the Hazmat in dangerous confined spaces (Fig. 6). Quince was developed for research into technical issues of UGVs at CBRNE disasters and has high mobility on rough terrain (Fig. 7).

Fukushima Daiichi Nuclear Power Plant Accident

At the Fukushima Daiichi nuclear power plant accident caused by tsunami in 2011, various disaster response robots were applied. They contributed to the cool shutdown and decommissioning of the plant. For example, PackBot and Quince gave essential data for task planning by shooting images and radiation measurement in nuclear reactor buildings there. Unmanned construction system removed debris outdoors that were contaminated by radiological materials and reduced the radiation rate there significantly.

Group INTRA in France and KHG in Germany are organizations for responding to nuclear plant accidents. They are equipped with robots and remote-controlled construction machines for radiation measurement, decontamination, and constructions in emergency. In Japan, the Assist Center for Nuclear Emergencies was established after the Fukushima Accident.
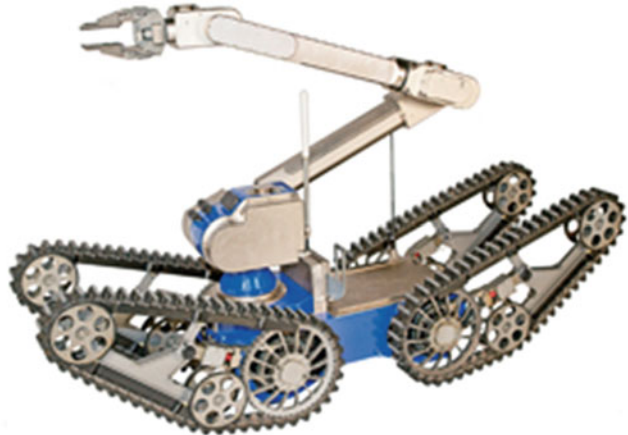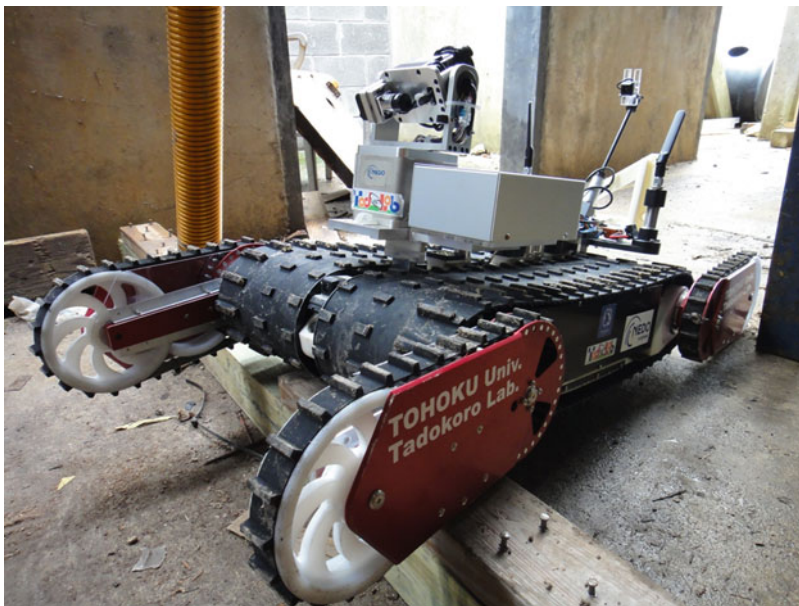

Disaster Response Robot, Fig. 5 Telemax (Courtesy of Cobham Mission Equipment) http://www.cobham.com/about-cobham/mission-systems/about-us/mission-equipment/unmanned-systems/products-and-services/remote-controlled-robotic-solutions/telemax-explosive-ordnance-(eod)-robot.aspx

**Disaster Response Robot, Fig. 6** PackBot (Courtesy of iRobot) http://www.irobot.com/us/learn/defense/packbot/Specifications.aspx

Disaster Response Robot, Fig. 7 Quince (Courtesy of International Rescue System Institute)

Summary and Future Directions

Data flow of disaster response robots is generally described by a feedback system as shown in Fig. 8. Robots change the states of objects and environment by movement and task execution. Sensors measure and recognize them, and their feedback enables the robots' autonomous motion and work. The sensed data are shown to operators via communication, data processing, and human interface. The operators give commands of motion and work to the system via the human interface. The system recognizes and transmits them to the robot.
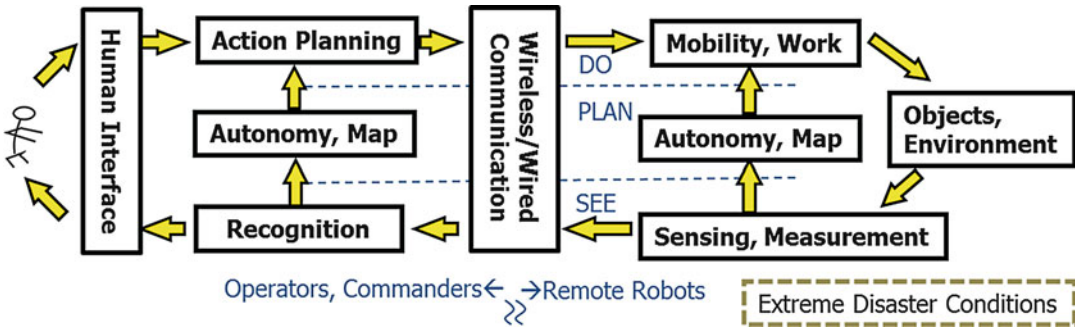
Disaster Response Robot, Fig. 8 Data flow of remotely controlled disaster response robots

Each functional block has its own technical challenges to be fulfilled under extreme environments of disaster space in response to the objectives and the conditions. They should be solved technically in order to improve the robot performance. They include insufficient mobility in disaster spaces (steps, gaps, slippage, narrow space, obstacles, etc.), deficient workability (dexterity, accuracy, speed, force, work space, etc.), poor sensors and sensor data processing (image, recognition, etc.), lack of reliability and performance of autonomy (robot intelligence, multiagent collaboration, etc.), issues of wireless and wired communication (instability, delay, capacity, tether handling, etc.), operators' limitations (situation awareness, decision ability, fatigue, mistake, etc.), basic performances (explosion proof, weight, durability, portability, etc.), and system integration that combines the components into the solution. Mission critical planning and execution including human factors, training, role sharing, logistics, etc. have to be considered at the same time.

Research into systems and control is expected to solve the abovementioned challenges of components and systems. For example, intelligent control is essential for mobility and workability under extreme conditions; control of feedback systems including long delay and dynamic instability, control of human-in-loop systems, and system integration of heterogeneous systems are important research topics of systems and control.

In the research field of disaster robotics, various competitions of practical robots have been held, e.g., RoboCupRescue targeting CBRNE disasters, ELROB and euRathlon for field activities, MAGIC for multi-robot autonomy, and DARPA Robotics Challenge for humanoid robots in nuclear disasters. These competitions seek to stimulate solutions of the above-mentioned technical issues in different environments by providing practical test beds for advanced technology developments.

Cross-References

▶ Robot Teleoperation
▶ Walking Robots
▶ Wheeled Robots

Bibliography

ELROB (2013) www.elrob.org
Group INTRA (2013) www.groupe-intra.com
KHG (2013) www.khgmbh.de
Murphy, R. (2014) Disaster robotics. MIT, Cambridge
RoboCup (2013) www.robocup.org
Siciliano B, Khatib O (eds) (2008) Springer handbook of robotics, 1st edn. Springer, Berlin
Siciliano B, Khatib O (eds) (2014) Springer handbook of robotics, 2nd edn. Springer, Berlin
Tadokoro S (ed) (2010) Rescue robotics: DDT project on robots and systems for urban search and rescue. Springer, London
Tadokoro S, Seki S, Asama H (2013) Priority issues of disaster robotics in Japan. In: Proceedings of the IEEE region 10 humanitarian technology conference, Sendai, 27–29 Aug 2013

Discrete Event Systems and Hybrid Systems, Connections Between

Alessandro Giua
DIEE, University of Cagliari, Cagliari, Italy
LSIS, Aix-en-Provence, France

Abstract

The causes of the complex behavior typical of hybrid systems are multifarious and are commonly explained in the literature using paradigms that are mainly focused on the connections between time-driven and hybrid systems. In this entry, we recall some of these paradigms and further explore the connections between discrete event and hybrid systems from other perspectives. In particular, the role of abstraction in passing from a hybrid model to a discrete event one and vice versa is discussed.

Keywords

Hybrid system; Logical discrete event system; Timed discrete event system

Introduction

Hybrid systems combine the dynamics of both *time-driven systems* and *discrete event systems*.

The evolution of a *time-driven system* can be described by a differential equation (in continuous time) or by a difference equation (in discrete time). An example of such a system is the tank shown in Fig. 1 whose behavior, assuming the tank is not full, is ruled in continuous time t by the differential equation

$$\frac{d}{dt}V(t) = q_1(t) - q_2(t)$$

where V is the volume of liquid and q_1 and q_2 are, respectively, the input and output flow.

A *discrete event system* (Lafortune and Cassandras 2007; Seatzu et al. 2012) evolves in accordance with the abrupt occurrence, at possibly unknown irregular intervals, of physical events. Its states may have logical or symbolic, rather than numerical, values that change in response to events which may also be described in nonnumerical terms. An example of such a system is a robot that loads parts on a conveyor, whose behavior is described by the automaton in Fig. 2. The robot can be "idle," "loading" a part, or in an "error" state when a part is incorrectly positioned. The events that drive its evolution are a (grasp a part), b (part correctly loaded), c (part incorrectly positioned), and d (part repositioned). In a *logical* discrete event system (▶ Supervisory Control of Discrete-Event Systems), the timing of event occurrences are ignored, while in a *timed* discrete event system (▶ Models for Discrete Event Systems: An Overview), they are described by means of a suitable timing structure.

In a *hybrid system* (▶ Hybrid Dynamical Systems, Feedback Control of), time-driven and event-driven evolutions are simultaneously present and mutually dependent. As an example, consider a room where a thermostat maintains the temperature $x(t)$ between $x_a = 20\,^\circ\text{C}$ and $x_b = 22\,^\circ\text{C}$ by turning a heat pump on and off. Due to the exchange with the external environment at temperature $x_e \ll x(t)$, when the pump is off, the room temperature derivative is

$$\frac{d}{dt}x(t) = -k[x(t) - x_e]$$

where k is a suitable coefficient, while when the pump is on, the room temperature derivative is

$$\frac{d}{dt}x(t) = h(t) - k[x(t) - x_e]$$

where the positive term $h(t)$ is due to the heat pump. The hybrid automaton that describes this system is shown in Fig. 3.

The causes of the complex behavior typical of hybrid systems are multifarious, and among the paradigms commonly used in the literature to describe them, we mention three.
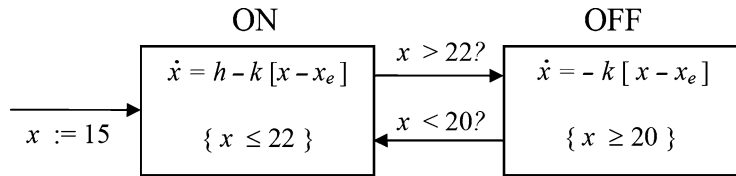
Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 1 A tank
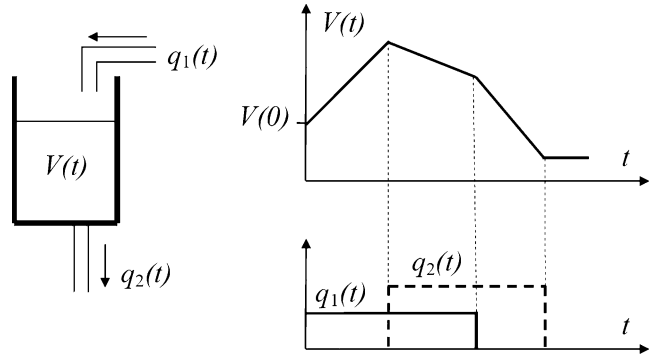
Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 2 A machine with failures
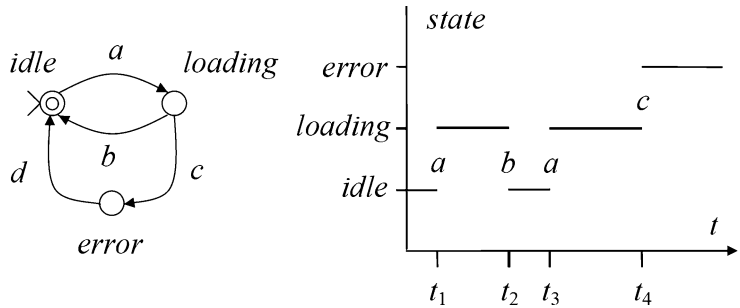
Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 3 Hybrid automaton of the thermostat

- *Logically controlled systems.* Often, a physical system with a time-driven evolution is controlled in a feedback loop by means of a controller that implements discrete computations and event-based logic. This is the case of the thermostat mentioned above. Classes of systems that can be described by this paradigm are *embedded systems* or, when the feedback loop is closed through a communication network, *cyber-physical systems*.
- *State-dependent mode of operation.* A time-driven system can have different modes of evolution depending on its current state. As an example, consider a bouncing ball. While the ball is above the ground (vertical position $h > 0$) its behavior is that of a falling body subject to a constant gravitational force. However, when the ball collides with the ground

(vertical position $h = 0$), its behavior is that of a (partially) elastic body that bounces up. Classes of systems that can be described by this paradigm are *piecewise affine systems* and *linear complementarity system*.
- *Variable structure systems.* Some systems may change their structure assuming different configuration, each characterized by a different behavior. As an example, consider a multicell voltage converter composed by a cascade of elementary commutation cells: controlling some switches, it is possible to insert or remove cells so as to produce a desired output voltage signal. Classes of systems that can be described by this paradigm are *switched systems*.

While these are certainly appropriate and meaningful paradigms, they are mainly focused on the connections between time-driven and

hybrid systems. In the rest of this entry, we will discuss the connections between discrete event and hybrid systems from other different perspectives. The focus is strictly on modeling, thus approaches for analysis or control will not be discussed.
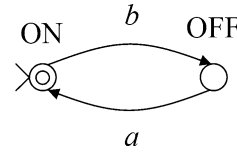
From Hybrid Systems to Discrete Event System by Modeling Abstraction

A *system* is a physical object, while a *model* is a (more or less accurate) mathematical description of its behavior that captures those features that are deemed mostly significant. In the previous pages, we have introduced different classes of systems, such as "time-driven systems," "discrete event systems," and "hybrid systems," but properly speaking, this taxonomy pertains to the models because the terms "time driven," "discrete event," or "hybrid" should be used to classify the mathematical description and not the physical object.

According to this view, a discrete event model is often perceived as a high-level description of a physical system where the time-driven dynamics are ignored or, at best, approximated by a timing structure. This procedure to derive a simpler model in a way that preserves the properties being analyzed while hiding the details that are of no interest is called *abstraction* (Alur et al. 2000).

Consider, as an example, the thermostat in Fig. 3. In such a system, the time-driven evolution determines a change in the temperature, which in turn – reaching a threshold – triggers the occurrence of an event that changes the discrete state. Assume one does not care about the exact form this triggering mechanism takes and is only interested in determining if the heat pump is turned on or off. In such a case, we can completely *abstract* the time-driven evolution obtaining a logical discrete event model such as the automaton in Fig. 4, where label *a* denotes the event *the temperature drops below* 20 °C and label *b* denotes the event *the temperature raises over* 22 °C.

For some purposes, e.g., to determine the utilization rate of the heat pump and thus its operating cost, the model in Fig. 4 is inadequate. In such a case, one can consider a less coarse

Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 4 Logical discrete event model of the thermostat

abstraction of the hybrid model in Fig. 3 obtaining a timed discrete event model such as the automaton in Fig. 4. Here, to each event is associated a firing delay: as an example, δ_a represents the time it takes – when the pump is off – to cool down until the lower temperature threshold is reached and event *a* occurs. The delay may be a deterministic value or even a random one to take into account the uncertainty due to non-modeled time-varying parameters such as the temperature of the external environment. Note that a new state (START) and a new event b' have now been introduced to capture the transient phase in which the room temperature, from the initial value $x(0) = 15$ °C, reaches the higher temperature threshold: in fact, event b' has a delay greater than the delay of event b.

Timed Discrete Event Systems Are Hybrid Systems

Properly speaking, all timed discrete event systems may also be seen as hybrid systems if one considers the dynamics of the timers – that specify the event occurrence – as elementary time-driven evolutions. In fact, the simplest model of hybrid systems is the *timed automaton* introduced by Alur and Dill (1994) whose main feature is the fact that each continuous variable $x(t)$ has a constant derivative $\dot{x}(t) = 1$ and thus can only describe the passage of time. Incidentally, we note that the term "timed automaton" is also used in the area of discrete event systems (Lafortune and Cassandras 2007) to denote an automaton in which a timing structure is associated to the events: such an example was shown in Fig. 5. To avoid any confusion, in the following, we denote the former model *Alur-Dill automaton* and the latter model *timed DES automaton*.

In Fig. 6 is shown an Alur-Dill automaton that describes the thermostat, where the time-driven dynamics have been abstracted and only the timing of event occurrence is modeled as in the timed DES automaton in Fig. 5. The only continuous variable is the value of a timer δ: when it goes beyond a certain threshold (e.g., $\delta > \delta_a$), an event occurs (e.g., event a) changing the discrete state (e.g., from OFF to ON) and resetting the timer to zero.

It is rather obvious that the behavior of the Alur-Dill automaton in Fig. 6 is equivalent to the behavior of timed DES automaton in Fig. 5. In the former model, the notion of time is encoded by means of an explicit continuous variable δ. In the latter model, the notion of time is implicitly encoded by the timer that during an evolution will be associated to each event. In both cases, however, the overall state of the systems is described by a pair $(\ell(t), x(t))$ where the first element ℓ takes value in a discrete set $\{START, ON, OFF\}$ and the second element is a vector (in this particular case with a single component) of timer valuations.

It should be pointed out that an Alur-Dill automaton may have a more complex structure than that shown in Fig. 6: as an example, the guard associated to a transition, i.e., the values of the timer that enable it, can be an arbitrary rectangular set. However, the same is also true for a timed discrete event system: several policies can be used to define the time intervals enabling an

event (enabling policy) or to specify when a timer is reset (memory policy) (Ajmone Marsan et al. 1995). Furthermore, timed discrete event system can have arbitrary stochastic timing structures (e.g., semi-Markovian processes, Markov chains, and queuing networks (Lafortune and Cassandras 2007)), not to mention the possibility of having an infinite discrete state space (e.g., timed Petri nets (Ajmone Marsan et al. 1995; David and Alla 2004)). As a result, we can say that timed DES automata are far more general than Alur-Dill automata and represent a meaningful subclass of hybrid systems.
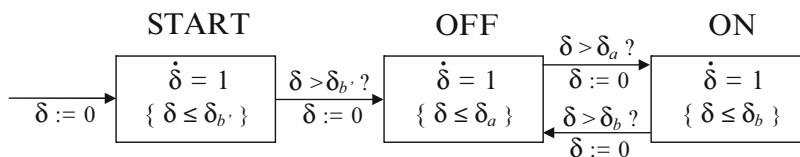
From Discrete Event System to Hybrid Systems by Fluidization

The computational complexity involved in the analysis and optimization of real-scale problems often becomes intractable with discrete event models due to the very large number of reachable states, and a technique that has shown to be effective in reducing this complexity is called *fluidization* (▶ Applications of Discrete-Event Systems). It should be noted that the derivation of a fluid (i.e., hybrid) model from a discrete event one is yet an example of abstraction albeit going in opposite direction with respect to the examples discussed in the section "From Hybrid Systems to Discrete Event System by Modeling Abstraction" above.

The main drive that motivated the fluidization approach derives from the observation that some discrete event systems are "heavily populated" in the sense that there are many identical items in some component (e.g., clients in a queue). Fluidization consists in replacing the integer counter of the number of items by a real number and in approximating the "fast" discrete event
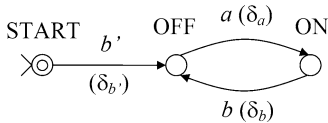
Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 5 Timed discrete event model of the thermostat

Discrete Event Systems and Hybrid Systems, Connections Between, Fig. 6 Alur-Dill automaton of the thermostat

dynamics that describe how the counter changes by a continuous dynamics. This approach has been successfully used to study the performance optimization of fluid-queuing networks (Cassandras and Lygeros 2006) or Petri net models (Balduzzi et al. 2000; David and Alla 2004; Silva and Recalde 2004) with applications in domains such as manufacturing systems and communication networks. We also remark that in general, different fluid approximations are necessary to describe the same system, depending on its discrete state, e.g., in the manufacturing domain, machines working or down, buffers full or empty, and so on. Thus, the resulting model can be better described as a hybrid model, where different time-driven dynamics are associated to different discrete states.

There are many advantages in using fluid approximations. First, there is the possibility of considerable increase in computational efficiency because the simulation of a fluid model can often be performed much faster than that of its discrete event counterpart. Second, fluid approximations provide an aggregate formulation to deal with complex systems, thus reducing the dimension of the state space. Third, the resulting simple structures often allow explicit computation of performance measures. Finally, some design parameters in fluid models are continuous; hence, it is possible to use gradient information to speed up optimization and to perform sensitivity analysis (Balduzzi et al. 2000): in many cases, it has also been shown that fluid approximations do not introduce significant errors when carrying out performance analysis via simulation (▸ Perturbation Analysis of Discrete Event Systems).

Cross-References

- ▸ Applications of Discrete-Event Systems
- ▸ Hybrid Dynamical Systems, Feedback Control of
- ▸ Models for Discrete Event Systems: An Overview
- ▸ Perturbation Analysis of Discrete Event Systems
- ▸ Supervisory Control of Discrete-Event Systems

Bibliography

Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalized stochastic Petri nets. Wiley, Chichester/New York

Alur R, Dill DL (1994) A theory of timed automata. Theor Comput Sci 126:183–235

Alur R, Henzinger TA, Lafferriere G, Pappas GJ (2000) Discrete abstractions of hybrid systems. Proc IEEE 88(7):971–984

Balduzzi F, Giua A, Menga G (2000) First–order hybrid Petri nets: a model for optimization and control. IEEE Trans Robot Autom 16:382–399

Cassandras CG, Lygeros J (eds) (2006) Stochastic hybrid systems: recent developments and research. CRC Press, New York

David R, Alla H (2004) Discrete, continuous and hybrid Petri nets. Springer, Berlin

Lafortune S, Cassandras CG (2007) Introduction to discrete event systems, 2nd edn. Springer, Boston

Seatzu C, Silva M, van Schuppen JH (eds) (2012) Control of discrete event systems. Automata and Petri net perspectives. Volume 433 of lecture notes in control and information science. Springer, London

Silva M, Recalde L (2004) On fluidification of Petri net models: from discrete to hybrid and continuous models. Annu Rev Control 28(2):253–266

Discrete Optimal Control

David Martin De Diego
Instituto de Ciencias Matemáticas
(CSIC-UAM-UC3M-UCM), Madrid, Spain

Synonyms

DOC

Abstract

Discrete optimal control is a branch of mathematics which studies optimization procedures for controlled discrete-time models – that is, the optimization of a performance index associated with a discrete-time control system. This entry gives an introduction to the topic. The formulation of a general discrete optimal control problem is described, and applications to mechanical systems are discussed.

Keywords

Discrete mechanics; Discrete-time models; Symplectic integrators; Variational integrators

Definition

Discrete optimal control is a branch of mathematics which studies optimization procedures for controlled discrete-time models, that is, the optimization of a performance index associated to a discrete-time control system.

Motivation

Optimal control theory is a mathematical discipline with innumerable applications in both science and engineering. Discrete optimal control is concerned with control optimization for discrete-time models. Recently, in discrete optimal control theory, a great interest has appeared in developing numerical methods to optimally control real mechanical systems, as for instance, autonomous robotic vehicles in natural environments such as robotic arms, spacecrafts, or underwater vehicles.

During the last years, a huge effort has been made for the comprehension of the fundamental geometric structures appearing in dynamical systems, including control systems and optimal control systems. This new geometric understanding of those systems has made possible the construction of suitable numerical techniques for integration. A collection of ad hoc numerical methods are available for both dynamical and control systems. These methods have grown up accordingly with the needs in research coming from different fields such as physics and engineering. However, a new breed of ideas in numerical analysis has started recently. They incorporate the geometry of the systems into the analysis and that allows faster and more accurate algorithms and with less spurious effects than the traditional ones. All this gives birth to a new field called Geometric Integration (Hairer et al. 2002). For instance, numerical integrators for Hamiltonian systems should preserve the symplectic

structure underlying the geometry of the system. If so, they are called symplectic integrators.

Another approach used by more and more authors is based on the theory of discrete mechanics and variational integrators to obtain geometric integrators preserving some of the geometry of the original system (Hussein et al. 2006; Marsden and West 2001; Wendlandt and Marsden 1997a,b) (see also the section "Discrete Mechanics"). These geometric integrators are easily adapted and applied to a wide range of mechanical systems: forced or dissipative systems, holonomically constrained systems, explicitly time-dependent systems, reduced systems with frictional contact, nonholonomic dynamics, and multisymplectic field theories, among others.

As before, in optimal control theory, it is necessary to distinguish two kinds of numerical methods: the so-called direct and indirect methods. If we use direct methods, we first discretize the state and control variables, control equations, and cost functional, and then we solve a nonlinear optimization problem with constraints given by the discrete control equations, additional constraints, and boundary conditions (Bock and Plitt 1984; Bonnans and Laurent-Varin 2006; Hager 2001; Pytlak 1999). In this case, we typically need to solve a system of the type (see the section "Formulation of a General Discrete Optimal Control Problem")

$$\begin{cases} \text{minimize } F(X) \ X = (q^0, \ldots, q^N, u_1, \ldots u_N) \\ \quad \text{with } \Psi(X) = 0 \\ \quad\quad \Phi(X) \geq 0 \end{cases}$$

On the other hand, indirect methods consist of solving numerically the boundary value problem obtained from the equations after applying Pontryagin's Maximum Principle.

The combination of direct methods and discrete mechanics allows to obtain numerical control algorithms which are geometric structure preserving and exhibit a good long-time behavior (Bloch et al. 2013; Jiménez et al. 2013; Junge and Ober-Blöbaum 2005; Junge et al. 2006; Kobilarov 2008; Leyendecker et al. 2007; Ober-Blöbaum 2008; Ober-Blöbaum et al. 2011). Furthermore, it is possible to adapt many of

the techniques used for continuous control mechanical systems to the design of quantitative and qualitative accurate numerical methods for optimal control methods (reduction by symmetries, preservation of geometric structures, Lie group methods, etc.).

Formulation of a General Discrete Optimal Control Problem

Let M be an n-dimensional manifold, x denote the state variables in M for an agent's environment, and $u \in U \subset \mathbb{R}^m$ be the control or action that the agent chooses to accomplish a task or objective. Let $f_d(x, u) \in M$ be the resulting state after applying the control u to the state x. For instance, x may be the configuration of a vehicle at time t and u its fuel consumption, and then $f_d(x, u)$ is the new configuration of the vehicle at time $t + h$, with $t, h > 0$. Of course, we want to minimize the fuel consumption. Hence, the optimal control problem consists of finding the cheapest way to move the system from a given initial position to a final state. The problem can be mathematically described as follows: find a sequence of controls $(u_0, u_1, \ldots, u_{N-1})$ and a sequence of states $(x_0, x_1, \ldots, x_N)$ such that

$$x_{k+1} = f_d(k, x_k, u_k), \qquad (1)$$

where $x_k \in M$, $u_k \in U$, and the total cost

$$\mathcal{C}_d = \sum_{k=0}^{N-1} C_d(k, x_k, u_k) + \phi_d(N, x_N) \qquad (2)$$

is minimized where ϕ_d is a function of the final time and state at the final time (the terminal payoff) and C_d is a function depending on the discrete time, the state, and the control at each intermediate discrete time k (the running payoff).

To solve the discrete optimal control problem determined by Eqs. (1) and (2), it is possible to use the classical Lagrangian multiplier approach. In this case, we consider the control equations (1) as constraint equations associating a Lagrange multiplier to each constraint. Assume for simplicity that $M = \mathbb{R}^n$. Then, we construct the augmented cost function

$$\tilde{\mathcal{C}}_d = \sum_{k=0}^{N-1} \Big[p_{k+1} \left(x_{k+1} - f_d(k, x_k, u_k) \right)$$
$$- C_d(k, x_k, u_k) \Big] - \Phi_d(N, x_N) \qquad (3)$$

where $p_k \in \mathbb{R}^n$, $k = 1, \ldots, N$, are considered as the Lagrange multipliers. The notation xy is used for the scalar (inner) product $x \cdot y$ of two vectors in $\mathbb{R}^n$.

From the *pseudo-Hamiltonian function*

$$H_d(k, x_k, p_{k+1}, u_k) = p_{k+1} f_d(k, x_k, u_k)$$
$$- C_d(k, x_k, u_k),$$

we deduce the necessary conditions for a constrained minimum:

$$x_{k+1} = \frac{\partial H_d}{\partial p}(k, x_k, p_{k+1}, u_k)$$
$$= f_d(k, x_k, u_k) \qquad (4)$$
$$p_k = \frac{\partial H_d}{\partial q}(k, x_k, p_{k+1}, u_k)$$
$$= p_{k+1} \frac{\partial f_d}{\partial q}(k, x_k, u_k)$$
$$- \frac{\partial C_d}{\partial q}(k, x_k, u_k) \qquad (5)$$
$$0 = \frac{\partial H_d}{\partial u}(k, x_k, p_{k+1}, u_k)$$
$$= p_{k+1} \frac{\partial f_d}{\partial u}(k, x_k, u_k)$$
$$- \frac{\partial C_d}{\partial u}(k, x_k, u_k) \qquad (6)$$

where $0 \le k \le N - 1$. Moreover, we have some boundary conditions

$$x_0 \text{ is given and } p_N = -\frac{\partial \Phi_d}{\partial q}(N, x_N) \quad (7)$$

The variable p_k is called the costate of the system and Eq. (5) is called the adjoint equation. Observe that the recursion of x_k given by Eq. (4) develops forward in the discrete time, but the recursion of the costate variable is backward in the discrete time.

In the sequel, it is assumed the following regularity condition:

$$\det\left(\frac{\partial^2 H_d}{\partial u^a \partial u^b}\right) \neq 0$$

where $1 \leq a, b \leq m$, and $(u^a) \in U \subseteq \mathbb{R}^m$. Applying the implicit function theorem, we obtain from Eq. (1) that locally $u_k = g(k, x_k, p_{k+1})$. Defining the function

$$\tilde{H}_d : \quad \mathbb{Z} \times \mathbb{R}^{2n} \longrightarrow \mathbb{R}$$
$$(k, q_k, p_{k+1}) \longmapsto H_d(k, q_k, p_{k+1}, u_k)$$

Equations (4) and (5) are rewritten as the following discrete Hamiltonian system:

$$x_{k+1} = \frac{\partial \tilde{H}_d}{\partial p}(k, x_k, p_{k+1}) \qquad (8)$$

$$p_k = \frac{\partial \tilde{H}_d}{\partial q}(k, x_k, p_{k+1}) \qquad (9)$$

The expression of the solutions of the optimal control problem as a discrete Hamiltonian system (under some regularity properties) is important since it indicates that the discrete evolution is preserving symplecticity. A simple proof of this fact is the following (de León et al. (2007)). Construct the following function

$$G_k(x_k, x_{k+1}, p_{k+1}) = \tilde{H}_d(k, x_k, p_{k+1})$$
$$- p_{k+1} x_{k+1},$$

with $0 \leq k \leq N - 1$. For each fixed k:

$$dG_k = \frac{\partial \tilde{H}_d}{\partial q}(k, x_k, p_{k+1})\, dx_k$$
$$+ \frac{\partial \tilde{H}_d}{\partial p}(k, x_k, p_{k+1})\, dp_{k+1}$$
$$- p_{k+1} dx_{k+1} - x_{k+1} dp_{k+1}\,.$$

Thus, along solutions of Eqs. (8) and (9), we have that $dG_{k|\text{solutions}} = p_k dx_k - p_{k+1} dx_{k+1}$ which implies $dx_k \wedge dp_k = dx_{k+1} \wedge dp_{k+1}$.

In the next section, we will study the case of discrete optimal control of mechanical systems.

First, we will need an introduction to discrete mechanics and variational integrators.

Discrete Mechanics

Let Q be an n-dimensional differentiable manifold with local coordinates (q^i), $1 \leq i \leq n$. We denote by TQ its tangent bundle with induced coordinates $(q^i, \dot{q}^i)$. Let $L: TQ \to \mathbb{R}$ be a Lagrangian function; the associated Euler–Lagrange equations are given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}^i}\right) - \frac{\partial L}{\partial q^i} = 0, \quad 1 \leq i \leq n. \quad (10)$$

These equations are a system of implicit second-order differential equations. Assume that the Lagrangian is **regular**, that is, the matrix $\left(\frac{\partial^2 L}{\partial \dot{q}^i \partial \dot{q}^j}\right)$ is non-singular. It is well known that the origin of these equations is variational (see Marsden and West 2001). Variational integrators retain this variational character and also some of the key geometric properties of the continuous system, such as symplecticity and momentum conservation (see Hairer et al. 2002 and references therein). In the following, we summarize the main features of this type of numerical integrators (Marsden and West 2001). A **discrete Lagrangian** is a map $L_d: Q \times Q \to \mathbb{R}$, which may be considered as an approximation of the integral action defined by a continuous Lagrangian $L: TQ \to \mathbb{R}: L_d(q_0, q_1) \approx \int_0^h L(q(t), \dot{q}(t))\, dt$ where $q(t)$ is a solution of the Euler–Lagrange equations for L with $q(0) = q_0$, $q(h) = q_1$, and $h > 0$ being enough small.

Remark 1 The Cartesian product $Q \times Q$ is equipped with an interesting differential structure, called Lie groupoid, which allows the extension of variational calculus to more general settings (see Marrero et al. 2006, 2010 for more details).

Define the **action sum** $S_d: Q^{N+1} \to \mathbb{R}$, corresponding to the Lagrangian L_d by $S_d = \sum_{k=1}^{N} L_d(q_{k-1}, q_k)$, where $q_k \in Q$ for $0 \leq k \leq N$ and N is the number of steps. The discrete variational principle states that the solutions of

the discrete system determined by L_d must extremize the action sum given fixed endpoints q_0 and q_N. By extremizing S_d over q_k, $1 \leq k \leq N-1$, we obtain the system of difference equations

$$D_1 L_d(q_k, q_{k+1}) + D_2 L_d(q_{k-1}, q_k) = 0, \quad (11)$$

or, in coordinates,

$$\frac{\partial L_d}{\partial x^i}(q_k, q_{k+1}) + \frac{\partial L_d}{\partial y^i}(q_{k-1}, q_k) = 0,$$

where $1 \leq i \leq n$, $1 \leq k \leq N-1$, and x, y denote the n-first and n-second variables of the function L_d, respectively.

These equations are usually called the **discrete Euler–Lagrange equations**. Under some regularity hypotheses (the matrix $D_{12}L_d(q_k, q_{k+1})$ is regular), it is possible to define a (local) discrete flow $\Upsilon_{L_d} \colon Q \times Q \to Q \times Q$, by $\Upsilon_{L_d}(q_{k-1}, q_k) = (q_k, q_{k+1})$ from (11). Define the discrete Legendre transformations associated to L_d as

$$\mathbb{F}^- L_d \colon Q \times Q \;\to\; T^*Q$$
$$(q_0, q_1) \longmapsto (q_0, -D_1 L_d(q_0, q_1)),$$
$$\mathbb{F}^+ L_d \colon Q \times Q \;\to\; T^*Q$$
$$(q_0, q_1) \longmapsto (q_1, D_2 L_d(q_0, q_1)),$$

and the discrete Poincaré–Cartan 2-form $\omega_d = (\mathbb{F}^+ L_d)^* \omega_Q = (\mathbb{F}^- L_d)^* \omega_Q$, where ω_Q is the canonical symplectic form on T^*Q. The discrete algorithm determined by Υ_{L_d} preserves the symplectic form ω_d, i.e., $\Upsilon_{L_d}^* \omega_d = \omega_d$. Moreover, if the discrete Lagrangian is invariant under the diagonal action of a Lie group G, then the discrete momentum map $J_d \colon Q \times Q \to \mathfrak{g}^*$ defined by

$$\langle J_d(q_k, q_{k+1}), \xi \rangle = \langle D_2 L_d(q_k, q_{k+1}),$$
$$\xi_Q(q_{k+1}) \rangle$$

is preserved by the discrete flow. Therefore, these integrators are symplectic-momentum preserving. Here, ξ_Q denotes the fundamental vector field determined by $\xi \in \mathfrak{g}$, where $\mathfrak{g}$ is the Lie algebra of G. As stated in Marsden and West (2001), discrete mechanics is inspired by discrete

formulations of optimal control problems (see Cadzow 1970; Hwang and Fan 1967; Jordan and Polak 1964).

Discrete Optimal Control of Mechanical Systems

Consider a mechanical system whose configuration space is an n-dimensional differentiable manifold Q and whose dynamics is determined by a Lagrangian $L \colon TQ \to \mathbb{R}$. The control forces are modeled as a mapping $f \colon TQ \times U \to T^*Q$, where $f(v_q, u) \in T_q^* Q$, $v_q \in T_q Q$ and $u \in U$, being U the control space. Observe that this last definition also covers configuration and velocity-dependent forces such as dissipation or friction (see Ober-Blöbaum et al. 2011).

The motion of the mechanical system is described by applying the principle of **Lagrange–D'Alembert**, which requires that the solutions $q(t) \in Q$ must satisfy

$$\delta \int_0^T L(q(t), \dot{q}(t)) \, dt$$
$$+ \int_0^T f(q(t), \dot{q}(t), u(t)) \, \delta q(t) \, dt = 0,$$
$$(12)$$

where $(q, \dot{q})$ are the local coordinates of TQ and where we consider arbitrary variations $\delta q(t) \in T_{q(t)}Q$ with $\delta q(0) = 0$ and $\delta q(T) = 0$ (since we are prescribing fixed initial and final conditions $(q(0), \dot{q}(0))$ and $(q(T), \dot{q}(T))$).

As we consider an optimal control problem, the forces f must be chosen, if they exist, as the ones that extremize the **cost functional**:

$$\int_0^T C(q(t), \dot{q}(t), u(t)) \, dt$$
$$+ \Phi(q(T), \dot{q}(T), u(T)), \quad (13)$$

where $C \colon TQ \times U \to \mathbb{R}$.

The optimal equations of motion can now be derived using Pontryagin's Maximum Principle. In general, it is not possible to explicitly integrate these equations. Then, it is necessary to apply a numerical method. In this work, using

discrete variational techniques, we first discretize the Lagrange–d'Alembert principle and then the cost functional. We obtain a numerical method that preserves some geometric features of the original continuous system as described in the sequel.

Discretization of the Lagrangian and Control Forces

To discretize this problem, we replace the tangent space TQ by the Cartesian product $Q \times Q$ and the continuous curves by sequences $q_0, q_1, \ldots q_N$ (we are using N steps, with time step h fixed, in such a way $t_k = kh$ and $Nh = T$). The discrete Lagrangian $L_d : Q \times Q \to \mathbb{R}$ is constructed as an approximation of the action integral in a single time step (see Marsden and West 2001), that is,

$$L_d(q_k, q_{k+1}) \approx \int_{kh}^{(k+1)h} L(q(t), \dot{q}(t)) \, dt.$$

We choose the following discretization for the external forces: $f_d^{\pm} : Q \times Q \times U \to T^*Q$, where $U \subset \mathbb{R}^m, \ m \leq n$, such that

$$f_d^-(q_k, q_{k+1}, u_k) \in T_{q_k}^* Q,$$
$$f_d^+(q_k, q_{k+1}, u_k) \in T_{q_{k+1}}^* Q.$$

f_d^+ and f_d^- are right and left discrete forces (see Ober-Blöbaum et al. 2011).

Discrete Lagrange–d'Alembert Principle

Given such forces, we define the discrete Lagrange–d'Alembert principle, which seeks sequences $\{q_k\}_{k=0}^N$ that satisfy

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1})$$
$$+ \sum_{k=0}^{N-1} \Big(f_d^-(q_k, q_{k+1}, u_k) \, \delta q_k$$
$$+ f_d^+(q_k, q_{k+1}, u_k) \, \delta q_{k+1} \Big) = 0,$$

for arbitrary variations $\{\delta q_k\}_{k=0}^N$ with $\delta q_0 = \delta q_N = 0$. After some straightforward

manipulations, we arrive to the forced discrete Euler–Lagrange equations

$$D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1})$$
$$+ f_d^+(q_{k-1}, q_k, u_{k-1})$$
$$+ f_d^-(q_k, q_{k+1}, u_k) = 0,$$
$$(14)$$

with $k = 1, \ldots, N - 1$.

Boundary Conditions

For simplicity, we assume that the boundary conditions of the continuous optimal control problem are given by $q(0) = x_0, \dot{q}(0) = v_0, q(T) = x_T, \dot{q}(T) = v_T$. To incorporate these conditions to the discrete setting, we use both the continuous and discrete Legendre transformations. From Marsden and West (2001), given a forced system, we can define the discrete momenta

$$\mu_k = -D_1 L_d(q_k, q_{k+1}) - f_d^-(q_k, q_{k+1}, u_k),$$
$$\mu_{k+1} = D_2 L_d(q_k, q_{k+1}) + f_d^+(q_k, q_{k+1}, u_k).$$

From the continuous Lagrangian, we have the momenta

$$p_0 = \mathbb{F}L(x_0, v_0) = (x_0, \frac{\partial L}{\partial v}(x_0, v_0))$$
$$p_T = \mathbb{F}L(x_T, v_T) = \left(x_T, \frac{\partial L}{\partial v}(x_T, v_T) \right)$$

Therefore, the natural choice of boundary conditions is

$$x_0 = q_0, \qquad x_T = q_N$$
$$\mathbb{F}L(x_0, v_0) = -D_1 L_d(q_0, q_1) - f_d^-(q_0, q_1, u_0)$$
$$\mathbb{F}L(x_T, v_T) = D_2 L_d(q_{N-1}, q_N)$$
$$+ f_d^+(q_{N-1}, q_N, u_{N-1})$$

that we add to the discrete optimal control problem.

Discrete Cost Function

We can also approximate the cost functional (13) in a single time step h by

$$
\begin{aligned}
&C_d(q_k, q_{k+1}, u_k) \\
&\approx \int_{kh}^{(k+1)h} C(q(t), \dot{q}(t), u(t))\, dt,
\end{aligned}
$$

yielding the **discrete cost functional**:

$$
\sum_{k=0}^{N-1} C_d(q_k, q_{k+1}, u_k) + \Phi_d(q_{N-1}, q_N, u_{N-1}).
$$

Discrete Optimal Control Problem

With all these elements, we have the following discrete optimal control problem:

$$
\begin{aligned}
\min \sum_{k=0}^{N-1} &C_d(q_k, q_{k+1}, u_k) \\
&+ \Phi_d(q_{N-1}, q_N, u_{N-1})
\end{aligned}
$$

subject to

$$
\begin{aligned}
&D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) \\
&+ f_d^+(q_{k-1}, q_k, u_{k-1}) + f_d^-(q_k, q_{k+1}, u_k) = 0, \\
&x_0 = q_0, \qquad x_T = q_N \\
&\frac{\partial L}{\partial v}(x_0, v_0) = -D_1 L_d(q_0, q_1) - f_d^-(q_0, q_1, u_0) \\
&\frac{\partial L}{\partial v}(x_T, v_T) = D_2 L_d(q_{N-1}, q_N) \\
&\qquad\qquad\quad + f_d^+(q_{N-1}, q_N, u_{N-1})
\end{aligned}
$$

with $k = 1, \ldots, N-1$ (see Jiménez and Martín de Diego 2010; Jiménez et al. 2013 for a modification of these equations admitting piecewise controls).

The system now is a constrained nonlinear optimization problem, that is, it corresponds to the minimization of a function subject to algebraic constraints. The necessary conditions for optimality are derived applying nonlinear programming optimization. For the concrete implementation, it is possible to use sequential quadratic programming (SQP) methods to numerically solve the nonlinear optimization problem (Ober-Blöbaum et al. 2011).

Optimal Control Systems with Symmetries

In many interesting cases, the continuous optimal control of a mechanical system is defined on a Lie group and the Lagrangian, cost function, control forces are invariant under the group action. The goal is again the same as in the previous section, that is, to move the system from its current state to a desired state in an optimal way. In this particular case, it is possible to adapt the contents of the section "Discrete Mechanics" in a similar way to the continuous case (from the standard Euler–Lagrange equations to the Euler–Poincaré equations) and to produce the so-called Lie group variational integrators. These methods preserve the Lie group structure avoiding the use of local charts, projections, or constraints. Based on these methods, for the case of controlled mechanical systems, we produce the discrete Euler–Poincaré equations with controls and the discrete cost function. Consequently, it is possible to deduce necessary optimality conditions for this class of invariant systems (see Bloch et al. 2009; Bou-Rabee and Marsden 2009; Hussein et al. 2006; Kobilarov and Marsden 2011).

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Numerical Methods for Nonlinear Optimal Control Problems
▶ Optimal Control and Mechanics
▶ Optimal Control and Pontryagin's Maximum Principle

Bibliography

Bock HG, Plitt KJ (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: 9th IFAC world congress, Budapest. Pergamon Press, pp 242–247

Bloch AM, Hussein I, Leok M, Sanyal AK (2009) Geometric structure-preserving optimal control of a rigid body. J Dyn Control Syst 15(3):307–330

Bloch AM, Crouch PE, Nordkvist N (2013) Continuous and discrete embedded optimal control problems and

their application to the analysis of Clebsch optimal control problems and mechanical systems. J Geom Mech 5(1):1–38

Bonnans JF, Laurent-Varin J (2006) Computation of order conditions for symplectic partitioned Runge-Kutta schemes with application to optimal control. Numer Math 103:1–10

Bou-Rabee N, Marsden JE (2009) Hamilton-Pontryagin integrators on Lie groups: introduction and structure-preserving properties. Found Comput Math 9(2):197–219

Cadzow JA (1970) Discrete calculus of variations. Int J Control 11:393–407

de León M, Martí'n de Diego D, Santamaría-Merino A (2007) Discrete variational integrators and optimal control theory. Adv Comput Math 26(1–3):251–268

Hager WW (2001) Numerical analysis in optimal control. In: International series of numerical mathematics, vol 139. Birkhäuser Verlag, Basel, pp 83–93

Hairer E, Lubich C, Wanner G (2002) Geometric numerical integration, structure-preserving algorithms for ordinary differential equations'. Springer series in computational mathematics, vol 31. Springer, Berlin

Hussein I, Leok M, Sanyal A, Bloch A (2006) A discrete variational integrator for optimal control problems on $SO(3)$'. In: Proceedings of the 45th IEEE conference on decision and control, San Diego, pp 6636–6641

Hwang CL, Fan LT (1967) A discrete version of Pontryagin's maximum principle. Oper Res 15:139–146

Jordan BW, Polak E (1964) Theory of a class of discrete optimal control systems. J Electron Control 17:697–711

Jiménez F, Martín de Diego D (2010) A geometric approach to Discrete mechanics for optimal control theory. In: Proceedings of the IEEE conference on decision and control, Atlanta, pp 5426–5431

Jiménez F, Kobilarov M, Martín de Diego D (2013) Discrete variational optimal control. J Nonlinear Sci 23(3):393–426

Junge O, Ober-Blöbaum S (2005) Optimal reconfiguration of formation flying satellites. In: IEEE conference on decision and control and European control conference ECC, Seville

Junge O, Marsden JE, Ober-Blöbaum S (2006) Optimal reconfiguration of formation flying spacecraft- a decentralized approach. In: IEEE conference on decision and control and European control conference ECC, San Diego, pp 5210–5215

Kobilarov M (2008) Discrete geometric motion control of autonomous vehicles. Thesis, Computer Science, University of Southern California

Kobilarov M, Marsden JE (2011) Discrete geometric optimal control on Lie groups. IEEE Trans Robot 27(4):641–655

Leok M (2004) Foundations of computational geometric mechanics, control and dynamical systems. Thesis, California Institute of Technology. Available in http://www.math.lsa.umich.edu/~mleok

Leyendecker S, Ober-Blobaum S, Marsden JE, Ortiz M (2007) Discrete mechanics and optimal control for constrained multibody dynamics. In: 6th international conference on multibody systems, nonlinear dynamics, and control, ASME international design engineering technical conferences, Las Vegas

Marrero JC, Martín de Diego D, Martínez E (2006) Discrete Lagrangian and Hamiltonian mechanics on Lie groupoids. Nonlinearity 19:1313–1348. Corrigendum: Nonlinearity 19

Marrero JC, Martín de Diego D, Stern A (2010) Lagrangian submanifolds and discrete constrained mechanics on Lie groupoids. Preprint, To appear in DCDS-A 35-1, January 2015.

Marsden JE, West M (2001) Discrete mechanics and variational integrators. Acta Numer 10: 357–514

Marsden JE, Pekarsky S, Shkoller S (1999a) Discrete Euler-Poincaré and Lie-Poisson equations. Nonlinearity 12:1647–1662

Marsden JE, Pekarsky S, Shkoller S (1999b) Symmetry reduction of discrete Lagrangian mechanics on Lie groups. J Geom Phys 36(1–2):140–151

Ober-Blöbaum S (2008) Discrete mechanics and optimal control. Ph.D. Thesis, University of Paderborn

Ober-Blöbaum S, Junge O, Marsden JE (2011) Discrete mechanics and optimal control: an analysis. ESAIM Control Optim Calc Var 17(2):322–352

Pytlak R (1999) Numerical methods for optimal control problems with state constraints. Lecture Notes in Mathematics, 1707. Springer-Verlag, Berlin, xvi+215 pp.

Wendlandt JM, Marsden JE (1997a) Mechanical integrators derived from a discrete variational principle. Phys D 106:2232–2246

Wendlandt JM, Marsden JE (1997b) Mechanical systems with symmetry, variational principles and integration algorithms. In: Alber M, Hu B, Rosenthal J (eds) Current and future directions in applied mathematics, (Notre Dame, IN, 1996), Birkhäuser Boston, Boston, MA, pp 219–261

Distributed Model Predictive Control

Gabriele Pannocchia
University of Pisa, Pisa, Italy

Abstract

Distributed model predictive control refers to a class of predictive control architectures in which a number of local controllers manipulate a subset of inputs to control a subset of outputs (states) composing the overall system. Different levels of communication and (non)cooperation exist, although in general the most compelling properties can be established only for cooperative schemes,

those in which all local controllers optimize local inputs to minimize the same plantwide objective function. Starting from state-feedback algorithms for constrained linear systems, extensions are discussed to cover output feedback, reference target tracking, and nonlinear systems. An outlook of future directions is finally presented.

Keywords

Constrained large-scale systems; Cooperative control systems; Interacting dynamical systems

Introduction and Motivations

Large-scale systems (e.g., industrial processing plants, power generation networks, etc.) usually comprise several interconnected units which may exchange material, energy, and information streams. The overall effectiveness and profitability of such large-scale systems depend strongly on the level of local effectiveness and profitability of each unit but also on the level of interactions among the different units. An overall optimization goal can be achieved by adopting a single *centralized* model predictive control (MPC) system (Rawlings and Mayne 2009) in which *all* control input trajectories are optimized simultaneously to minimize a *common* objective.

This choice is often avoided for several reasons. When the overall number of inputs and states is very large, a single optimization problem may require computational resources (CPU time, memory, etc.) that are not available and/or compatible with the system's dynamics. Even if these limitations do no hold, it is often the case that organizational reasons require the use of smaller, local controllers, which are easier to coordinate and maintain.

Thus, industrial control systems are often *decentralized*, i.e., the overall system is divided into (possibly mildly coupled) subsystems and a local controller is designed for each unit disregarding the interactions from/to other subsystems. Depending on the extent of dynamic coupling, it is well known that the performance of such decentralized systems may be poor, and stability

properties may be even lost. *Distributed* predictive control architectures arise to meet performance specifications (stability at minimum) similar to centralized predictive control systems, still retaining the modularity and local character of the optimization problems solved by each controller.

Definitions and Architectures for Constrained Linear Systems

Subsystem Dynamics, Constraints, and Objectives

We start the description of distributed MPC algorithms by considering an overall discrete-time linear time-invariant system in the form:

$$x^+ = Ax + Bu, \quad y = Cx \qquad (1)$$

in which $x \in \mathbb{R}^n$ and $x^+ \in \mathbb{R}^n$ are, respectively, the system state at a given time and at a successor time: $u \in \mathbb{R}^m$ is the input: and $y \in \mathbb{R}^p$ is the output.

We consider that the overall system (1) is divided into M subsystems, $\mathbb{S}_i$, defined by (disjoint) sets of inputs and outputs (states), and each $\mathbb{S}_i$ is regulated by a *local* MPC. For each $\mathbb{S}_i$, we denote by $y_i \in \mathbb{R}^{p_i}$ its output, by $x_i \in \mathbb{R}^{n_i}$ its state, and by $u_i \in \mathbb{R}^{m_i}$ the control input computed by the ith MPC. Due to interactions among subsystems, the local output y_i (and state x_i) is affected by control inputs computed by (some) other MPCs. Hence, the dynamics of $\mathbb{S}_i$ can be written as

$$x_i^+ = A_i x_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j, \quad y_i = C_i x_i \tag{2}$$

in which $\mathcal{N}_i$ denotes the indices of *neighbors* of $\mathbb{S}_i$, i.e., the subsystems whose inputs have an influence on the states of $\mathbb{S}_i$. To clarify the notation, we depict in Fig. 1 the case of three subsystems, with neighbors $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1\}$, and $\mathcal{N}_3 = \{2\}$.

Without loss of generality, we assume that each pair (A_i, B_i) is stabilizable. Moreover, the state of each subsystem x_i is assumed known (to the ith MPC) at each decision time. For each subsystem $\mathbb{S}_i$, inputs are required to fulfill (hard) constraints:
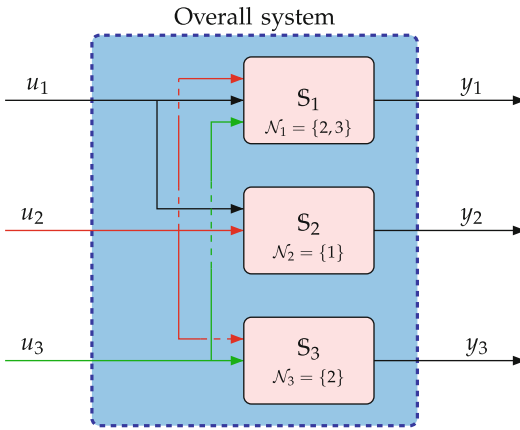
Distributed Model Predictive Control, Fig. 1
Interconnected systems and neighbors definition

$$u_i \in \mathbb{U}_i, \qquad i = 1, \ldots, M \qquad (3)$$

in which $\mathbb{U}_i$ are polyhedrons containing the origin in their interior. Moreover, we consider a quadratic stage cost function $\ell_i(x, u) \triangleq \frac{1}{2}(x'Q_i x + u'R_i u)$ and a terminal cost function $V_{fi}(x) \triangleq \frac{1}{2}x'P_i x$, with $Q_i \in \mathbb{R}^{n_i \times n_i}$, $R_i \in \mathbb{R}^{m_i \times m_i}$, and $P_i \in \mathbb{R}^{n_i \times n_i}$ positive definite. Without loss of generality, let $x_i(0)$ be the state of $\mathbb{S}_i$ at the current decision time. Consequently, the finite-horizon cost function associated with $\mathbb{S}_i$ is given by:

$$V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) \triangleq \sum_{i=0}^{N-1} \ell_i(x_i(k), u_i(k))$$
$$+ V_{fi}(x_i(N)) \qquad (4)$$

in which $\mathbf{u}_i = (u_i(0), u_i(1), \ldots, u_i(N-1))$ is a finite-horizon sequence of control inputs of $\mathbb{S}_i$, and $\mathbf{u}_j$ is similarly defined as a sequence of control inputs of each neighbor $j \in \mathcal{N}_i$. Notice that $V_i(\cdot)$ is a function of neighbors' input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, due to the dynamics (2).

Decentralized, Noncooperative, and Cooperative Predictive Control Architectures

Several levels of communications and (non) cooperation can exist among the controllers, as depicted in Fig. 2 for the case of two subsystems.

In *decentralized* MPC architectures, interactions among subsystems are neglected by forcing $\mathcal{N}_i = \varnothing$ for all i even if this is not true. That is, the subsystem model used in each local controller, instead of (2), is simply

$$x_i^+ = A_i x_i + B_i u_i, \, y_i = C_i x_i \qquad (5)$$

Therefore, an inherent mismatch exists between the model used by the local controllers (5) and the actual subsystem dynamics (2). Each local MPC solves the following finite-horizon optimal control problem (FHOCP):
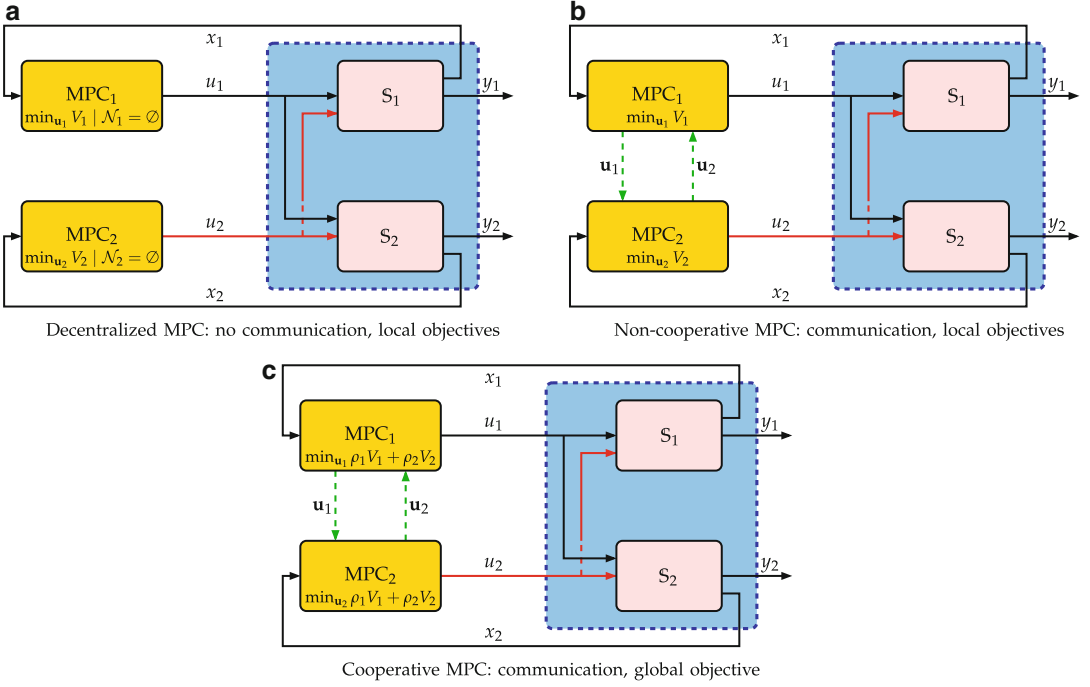
$$\mathbb{P}_i^{\text{De}} : \min_{\mathbf{u}_i} V_i(\cdot) \quad \text{s.t. } \mathbf{u}_i \in \mathbb{U}_i^N, \mathcal{N}_i = \varnothing \qquad (6)$$

We observe that in this case, $V_i(\cdot)$ *depends only* on local inputs, $\mathbf{u}_i$, because it is assumed that $\mathcal{N}_i = \varnothing$. Hence, each $\mathbb{P}_i^{\text{De}}$ is solved independently of the neighbors computations, and no iterations are performed. Clearly, depending on the actual level of interactions among subsystems, decentralized MPC architectures can perform poorly, namely, being non-stabilizing. Performance certifications are still possible resorting to robust stability theory, i.e., by treating the neglected dynamics $\sum_{j \in \mathcal{N}_i} B_{ij} u_j$ as (bounded) disturbances (Riverso et al. 2013).

In *noncooperative* MPC architectures, the existing interactions among the subsystems are fully taken into account through (2). Given a known value of the neighbors' control input sequences, $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$, each local MPC solves the following FHOCP:

$$\mathbb{P}_i^{\text{NCDi}} : \quad \min_{\mathbf{u}_i} V_i(\cdot) \text{ s.t. } \mathbf{u}_i \in \mathbb{U}_i^N \qquad (7)$$

The obtained solution can be exchanged with the other local controllers to update the assumed neighbors' control input sequences, and iterations can be performed. We observe that this approach is noncooperative because local controllers try to optimize different, possibly competing, objectives. In general, no convergence is guaranteed in noncooperative iterations, and when this scheme converges, it leads to a so-called Nash equilibrium. However, the achieved local control inputs do not have proven stability properties (Rawlings

a
Decentralized MPC: no communication, local objectives

b
Non-cooperative MPC: communication, local objectives

c
Cooperative MPC: communication, global objective

Distributed Model Predictive Control, Fig. 2 Three distributed control architectures: decentralized MPC, noncooperative MPC, and cooperative MPC

and Mayne 2009, §6.2.3). To ensure closed-loop stability, variants can be formulated by including a sequential solution of local MPC problems, exploiting the notion (if any) of an auxiliary stabilizing decentralized control law non-iterative noncooperative schemes are also proposed, in which stability guarantees are provided by ensuring a decrease of a centralized Lyapunov function at each decision time.

Finally, in *cooperative* MPC architectures, each local controller optimizes a common (plantwide) objective:

$$V(x(0), \mathbf{u}) \stackrel{\Delta}{=} \sum_{i=1}^{M} \rho_i V_i(x_i(0), \mathbf{u}_i, \{\mathbf{u}_j\}_{j \in \mathcal{N}_i}) \tag{8}$$

in which $\rho_i > 0$, for all i, are given scalar weights and $\mathbf{u} \stackrel{\Delta}{=} (\mathbf{u}_1, \ldots, \mathbf{u}_M)$ is the overall control sequence. In particular, given a known value of other subsystems' control input sequences, $\{\mathbf{u}_j\}_{j \neq i}$, each local MPC solves the following FHOCP:

$$\mathbb{P}_i^{\text{CDi}} : \quad \min_{\mathbf{u}_i} V(\cdot) \quad \text{s.t.} \ \mathbf{u}_i \in \mathbb{U}_i^N \tag{9}$$

As in noncooperative schemes, the obtained solution can be exchanged with the other local controllers, and further iterations can be performed. Notice that in $\mathbb{P}_i^{\text{CDi}}$, the (possible) implications of the local control sequence $\mathbf{u}_i$ to all other subsystems' objectives, $V_j(\cdot)$ with $j \neq i$ are taken into account, as well as the effect of the neighbors' sequences $\{\mathbf{u}_j\}_{j \in \mathcal{N}_i}$ on the local state evolution through (2). Clearly, this approach is termed cooperative because all controllers compute *local* inputs to minimize a *global* objective. Convergence of cooperative iterations is guaranteed, and under suitable assumptions the converged solution is the centralized Pareto-optimal solution (Rawlings and Mayne 2009, §6.2.4). Furthermore, the achieved local control inputs have proven stabilizing properties (Stewart et al. 2010). Variants are also proposed in which each controller still optimizes a local objective, but cooperative iterations are performed to ensure a

decrease of the global objective at each decision time (Maestre et al. 2011).

Cooperative Distributed MPC

Cooperative schemes are preferable over noncooperative schemes from many points of view, namely, in terms of superior theoretical guarantees and no larger computational requirements. In this section we focus on a prototype cooperative distributed MPC algorithm adapted from Stewart et al. (2010), highlighting the required computations and discussing the associated theoretical properties and guarantees.

Basic Algorithm
We present in Algorithm 1 a streamlined description of a cooperative distributed MPC algorithm, in which each local controller solves $\mathbb{P}_i^{\text{CDi}}$, given a previously computed value of all other subsystems' input sequences. For each local controller, the new iterate is defined as a convex combination of the newly computed solution with the previous iteration. A relative tolerance is defined, so that cooperative iterations stop when all local controllers have computed a new iterate sufficiently close to the previous one. A maximum number of cooperative iterations can also be defined, so that a finite bound on the execution time can be established.

Algorithm 1 *(Cooperative MPC).* **Require:** Overall warm start $\mathbf{u}^0 \triangleq (\mathbf{u}_1^0, \ldots, \mathbf{u}_M^0)$, convex step weights $w_i > 0$, s.t. $\sum_{i=1}^M w_i = 1$, relative tolerance parameter $\varepsilon > 0$, maximum cooperative iterations $c_{\max}$

1: Initialize: $c \leftarrow 0$ and $e_i \leftarrow 2\epsilon$ for $i = 1, \ldots, M$.
2: **while** $(c < c_{\max})$ **and** $(\exists i \,|\, e_i > \epsilon)$ **do**
3:   $c \leftarrow c + 1$.
4:   **for** $i = 1$ **to** M **do**
5:     Solve $\mathbb{P}_i^{\text{CDi}}$ in (9) obtaining $\mathbf{u}_i^*$.
6:   **end for**
7:   **for** $i = 1$ **to** M **do**
8:     Define new iterate: $\mathbf{u}_i^c \triangleq w_i \mathbf{u}_i^* + (1 - w_i) \mathbf{u}_i^{c-1}$.
9:     Compute convergence error: $e_i \triangleq \frac{\|\mathbf{u}_i^c - \mathbf{u}_i^{c-1}\|}{\|\mathbf{u}_i^{c-1}\|}$.
10:   **end for**
11: **end while**
12: **return** Overall solution: $\mathbf{u}^c \triangleq (\mathbf{u}_1^c, \ldots, \mathbf{u}_M^c)$.

We observe that Step 8 implicitly defines the new overall iterate as a convex combination of the overall solutions *achieved* by each controller, that is,

$$\mathbf{u}^c = \sum_{i=1}^M w_i (\mathbf{u}_1^{c-1}, \ldots, \mathbf{u}_i^*, \ldots, \mathbf{u}_M^{c-1}) \quad (10)$$

It is also important to observe that Steps 5, 8, and 9 are performed separately by each controller.

Properties
The basic cooperative MPC described in Algorithm 1 enjoys several nice theoretical and practical properties, as detailed (Rawlings and Mayne 2009, §6.3.1):
1. *Feasibility of each iterate*: $\mathbf{u}_i^{c-1} \in \mathbb{U}_i^N$ implies $\mathbf{u}_i^c \in \mathbb{U}_i^N$, for all $i = 1, \ldots, M$ and $c \in \mathbb{I}_{>0}$.
2. *Cost decrease at each iteration*: $V(x(0), \mathbf{u}^c) \leq V(x(0), \mathbf{u}^{c-1})$ for all $c \in \mathbb{I}_{>0}$.
3. *Cost convergence to the centralized optimum*: $\lim_{c \to \infty} V(x(0), \mathbf{u}^c) = \min_{\mathbf{u} \in \mathbb{U}^N} V(x(0), u)$, in which $\mathbb{U} \triangleq \mathbb{U}_1 \times \cdots \times \mathbb{U}_M$.

Resorting to suboptimal MPC theory, the above properties (1) and (2) can be exploited to show that the origin of closed-loop system

$$x^+ = Ax + B\kappa^c(x), \text{ with } \kappa^c(x) \triangleq u^c(0) \quad (11)$$

is *exponentially stable* for *any finite* $c \in \mathbb{I}_{>0}$. This result is of paramount (practical and theoretical) importance because it ensures closed-loop stability using cooperative distributed MPC with any finite number of cooperative iterations. As in centralized MPC based on the solution of a FHOCP (Rawlings and Mayne 2009, §2.4.3), particular care of the terminal cost function $V_{fi}(\cdot)$ is necessary, possibly in conjunction with a terminal constraint $x_i(N) \in \mathbb{X}_{fi}$. Several options can be adopted as discussed, e.g., in Stewart et al. (2010, 2011).

Moreover, the results in Pannocchia et al. (2011) can be used to show *inherent robust stability* to system's disturbances and measurement errors. Therefore, we can confidently

state that well-designed distributed cooperative MPC and centralized MPC algorithms share the *same guarantees* in terms of stability and robustness.

Complementary Aspects

We discuss in this section a number of complementary aspects of distributed MPC algorithms, omitting technical details for the sake of space.

Coupled Input Constraints and State Constraints

Convergence of the solution of cooperative distributed MPC towards the centralized (global) optimum holds when input constraints are in the form of (3), i.e., when no constraints involve inputs of different subsystems. Sometimes this assumption fails to hold, e.g., when several units share a common utility resource, that is, in addition to (3) some constraints involve inputs of more than one unit. In this situation, it is possible that Algorithm 1 remains stuck at a fixed point, without improving the cost, even if it is still away from the centralized optimum (Rawlings and Mayne 2009, §6.3.2). It is important to point out that this situation is harmless from a closed-loop stability and robustness point of view. However, the degree of suboptimality in comparison with centralized MPC could be undesired from a performance point of view. To overcome this situation, a slightly different partitioning of the overall inputs into non-disjoint sets can be adopted (Stewart et al. 2010).

Similarly the presence of state constraints, even in decentralized form $x_i \in \mathbb{X}_i$ (with $i = 1, \ldots, M$), can prevent convergence of a cooperative algorithm towards the centralized optimum. It is also important to point out that the local MPC controlling $\mathbb{S}_i$ needs to consider in the optimal control problem, besides local state constraints $x_i \in \mathbb{X}_i$, also state constraints of all other subsystems $\mathbb{S}_j$ such that $i \in \mathcal{N}_j$. This ensures feasibility of each iterate and cost reduction, hence closed-loop stability (and robustness) can be established.

Output Feedback and Offset-Free Tracking

When the subsystem state cannot be directly measured, each local controller can use a local state estimator, namely, a Kalman filter (or Luenberger observer). Assuming that the pair (A_i, C_i) is detectable, the subsystem state estimate evolves as follows:

$$\hat{x}_i^+ = A_i \hat{x}_i + B_i u_i + \sum_{j \in \mathcal{N}_i} B_{ij} u_j + L_i(y_i - C_i \hat{x}_i) \tag{12}$$

in which $L_i \in \mathbb{R}^{n_i \times p_i}$ is the local Kalman predictor gain, chosen such that the matrix $(A_i - L_i C_i)$ is Schur. Stability of the closed-loop origin can be still established using minor variations (Rawlings and Mayne 2009, §6.3.3).

When offset-free control is sought, each local MPC can be equipped with an integrating disturbance model similarly to centralized offset-free MPC algorithms (Pannocchia and Rawlings 2003). Given the current estimate of the subsystem state and disturbance, a target calculation problem is solved to compute the state and input equilibrium pair such that (a subset of) output variables correspond to given set points. Such a target calculation problem can be performed in a centralized fashion or in a distributed manner, although in the latter case several issues arise and associated precautions should be taken into account (Rawlings and Mayne 2009, §6.3.4).

Distributed Control for Nonlinear Systems

Several nonlinear distributed MPC algorithms have been recently proposed (Liu et al. 2009; Stewart et al. 2011). Some schemes require the presence of a coordinator, thus introducing a hierarchical structure (Scattolini 2009). In Stewart et al. (2011), instead, a cooperative distributed MPC architecture similar to the one discussed in the previous section has been proposed for nonlinear systems. Each local controller considers the following subsystem model:

$$\begin{aligned} x_i^+ &= f_i(x_i, u_i, u_j), \quad \text{with } j \in \mathcal{N}_i \\ y_i &= h_i(x_i) \end{aligned} \tag{13}$$

A problem (formally) identical to $\mathbb{P}_i^{\text{CDi}}$ in (9) is solved by each controller and cooperative iterations are performed. However, non-convexity of $\mathbb{P}_i^{\text{CDi}}$ can make a convex combination step similar to Step 8 in Algorithm 1 not necessarily a cost improvement. As a workaround in such cases, Stewart et al. (2011) propose deleting the least effective control sequence computed by a local controller (repeating this deletion if necessary). In this way it is possible to show a monotonic decrease of the cost function at each cooperative iteration.

Summary and Future Directions

We presented the basics and foundations of distributed model predictive control (DMPC) schemes, which prove useful and effective in the control of large-scale systems for which a single centralized predictive controller is not regarded as a possible or desirable solution, e.g., due to organizational requirements and/or computational limitations. In DMPCs, the overall controlled system is organized into a number of subsystems, in general featuring some dynamic couplings, and for each subsystem a local MPC is implemented.

Different flavors of communication and cooperation among the local controllers can be chosen by the designer, ranging from *decentralized* to *cooperative* schemes. In cooperative DMPC algorithms, the dynamic interactions among the subsystems are fully taken into account, with limited communication overheads, and the same overall objective can be optimized by each local controller. When cooperative iterations are performed upon convergence, such DMPC algorithms achieve the same global minimum control sequence as that of the centralized MPC. Termination prior to convergence does not hinder stability and robustness guarantees.

In this contribution, after discussing an overview on possible communication and cooperation schemes, we addressed the design of a state-feedback and distributed MPC algorithm for linear systems subject to input constraints, with convergence and stability guarantees. Then,

we discussed various extensions to coupled input constraints and state constraints, output feedback, reference target tracking, and nonlinear systems.

The research on DMPC algorithms has been extensive during the last decade, and some excellent review papers have been recently made available (Christofides et al. 2013; Scattolini 2009). Still, we expect DMPC to attract research efforts in various directions, as briefly discussed:

- *Nonlinear DMPC* algorithms (Liu et al. 2009; Stewart et al. 2011) will require improvements in terms of global optimum goals.
- *Economic DMPC* and *tracking DMPC* (Ferramosca et al. 2013) will replace current formulations designed for regulation around the origin, especially for nonlinear systems.
- *Reconfigurability*, e.g., addition/deletion of new local controllers, is an ongoing topic, and preliminary results available for decentralized architectures (Riverso et al. 2013) may be extended to cooperative and noncooperative schemes. It is also desirable to improve the resilience of DMPC to *communication disruptions* (Alessio et al. 2011).
- Preliminary results on *constrained distributed estimation* (Farina et al. 2012) will draw attention and require further insights to bridge the gap between constrained estimation and control algorithms.
- Specific *optimization algorithms* tailored to DMPC local problems (Doan et al. 2011) will increase the effectiveness of DMPC algorithms, as well as *distributed optimization* approaches will be exploited even for dynamically uncoupled systems.

Cross-References

Recommended Reading

General overviews on DMPC can be found in Christofides et al. (2013), Rawlings and Mayne (2009), and Scattolini (2009). DMPC algorithms for linear systems are discussed in Alessio et al. (2011), Ferramosca et al. (2013), Riverso et al. (2013), Stewart et al. (2010), and Maestre et al. (2011), and for nonlinear systems in Farina et al. (2012), Liu et al. (2009), and Stewart et al. (2011). Supporting results for implementation and robustness theory can be found in Doan et al. (2011), Pannocchia and Rawlings (2003), and Pannocchia et al. (2011).

Bibliography

Alessio A, Barcelli D, Bemporad A (2011) Decentralized model predictive control of dynamically coupled linear systems. J Process Control 21:705–714

Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J (2013) Distributed model predictive control: a tutorial review and future research directions. Comput Chem Eng 51:21–41

Doan MD, Keviczky T, De Schutter B (2011) An iterative scheme for distributed model predictive control using Fenchel's duality. J Process Control 21:746–755

Farina M, Ferrari-Trecate G, Scattolini R (2012) Distributed moving horizon estimation for nonlinear constrained systems. Int J Robust Nonlinear Control 22:123–143

Ferramosca A, Limon D, Alvarado I, Camacho EF (2013) Cooperative distributed MPC for tracking. Automatica 49:906–914

Liu J, Muñoz de la Peña D, Christofides PD (2009) Distributed model predictive control of nonlinear process systems. AIChE J 55(5):1171–1184

Maestre JM, Muñoz de la Peña D, Camacho EF (2011) Distributed model predictive control based on a cooperative game. Optim Control Appl Methods 32:153–176

Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model predictive control. AIChE J 49:426–437

Pannocchia G, Rawlings JB, Wright SJ (2011) Conditions under which suboptimal nonlinear MPC is inherently robust. Syst Control Lett 60:747–755

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison

Riverso S, Farina M, Ferrari-Trecate G (2013) Plug-and-play decentralized model predictive control for linear systems. IEEE Trans Autom Control 58:2608–2614

Scattolini R (2009) A survey on hierarchical and distributed model predictive control. J Process Control 19:723–731

Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G (2010) Cooperative distributed model predictive control. Syst Control Lett 59:460–469

Stewart BT, Wright SJ, Rawlings JB (2011) Cooperative distributed model predictive control for nonlinear systems. J Process Control 21:698–704

Distributed Optimization

Angelia Nedić
Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL, USA

Abstract

The paper provides an overview of the distributed first-order optimization methods for solving a constrained convex minimization problem, where the objective function is the sum of local objective functions of the agents in a network. This problem has gained a lot of interest due to its emergence in many applications in distributed control and coordination of autonomous agents and distributed estimation and signal processing in wireless networks.

Keywords

Collaborative multi-agent systems; Consensus protocol; Gradient-projection method; Networked systems

Introduction

There has been much recent interest in distributed optimization pertinent to optimization aspects arising in control and coordination of networks consisting of multiple (possibly mobile) agents and in estimation and signal processing in sensor networks (Bullo et al. 2009; Hendrickx 2008; Kar and Moura 2011; Martinoli et al. 2013; Mesbahi and Egerstedt 2010; Olshevsky 2010). In many of these applications, the network system goal is to optimize a global objective

through local agent-based computations and local information exchange with immediate neighbors in the underlying communication network. This is motivated mainly by the emergence of large-scale data and/or large-scale networks and new networking applications such as mobile ad hoc networks and wireless sensor networks, characterized by the lack of centralized access to information and time-varying connectivity. Control and optimization algorithms deployed in such networks should be completely distributed (relying only on local observations and information), robust against unexpected changes in topology (i.e., link or node failures) and against unreliable communication (noisy links or quantized data) (see ▶ Networked Systems). Furthermore, it is desired that the algorithms are scalable in the size of the network.

Generally speaking, the problem of distributed optimization consists of three main components:

1. The optimization problem that the network of agents wants to solve collectively (specifying an objective function and constraints)
2. The local information structure, which describes what information is locally known or observable by each agent in the system (who knows what and when)
3. The communication structure, which specifies the connectivity topology of the underlying communication network and other features of the communication environment

The algorithms for solving such global network problems need to comply with the distributed knowledge about the problem among the agents and obey the local connectivity structure of the communication network (▶ Networked Systems; ▶ Graphs for Modeling Networked Interactions).

Networked System Problem

Given a set $N = \{1, 2, \ldots, n\}$ of agents (also referred to as nodes), the global system problem has the following form:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} f_i(x) \\
\text{subject to} \quad & x \in X.
\end{aligned}
\tag{1}
$$

Each $f_i : \mathrm{R}^d \to \mathrm{R}$ is a convex function which represents the local objective of agent i, while $X \subseteq \mathrm{R}^d$ is a closed convex set. The function f_i is a private function known only to agent i, while the set X is commonly known by all agents $i \in N$. The vector $x \in X$ represents a global decision vector which the agents want to optimize using local information. The problem is a simple constrained convex optimization problem, where the global objective function is given by the sum of the individual objective functions $f_i(x)$ of the agents in the system. As such, the objective function is the sum of non-separable convex functions corresponding to multiple agents connected over a network.

As an example, consider the problem arising in support vector machines (SVMs), which are a popular tool for classification problems. Each agent i has a set $S_i = \left\{ \left(a_j^{(i)}, b_j^{(i)} \right) \right\}_{j=1}^{m_i}$ of m_i sample-label pairs, where $a_j^{(i)} \in \mathrm{R}^d$ is a data point and $b_j^{(i)} \in \{+1, -1\}$ is its corresponding (correct) label. The number m_i of data points for every agent i is typically very large (hundreds of thousands). Without sharing the data points, the agents want to collectively find a hyperplane that separates all the data, i.e., a hyperplane that separates (with a maximal separation distance) the data with label 1 from the data with label -1 in the global data set $\bigcup_{i=1}^{n} S_i$. Thus, the agents need to solve an unconstrained version of the problem (1), where the decision variable $x \in \mathrm{R}^d$ is a hyperplane normal and the objective function f_i of agent i is given by

$$
f_i(x) = \frac{\lambda}{2} \|x\|^2 + \sum_{j=1}^{m_i} \max \left\{ 0, 1 - b_j^{(i)} \left(x' a_j^{(i)} \right) \right\},
$$

where λ is a regularization parameter (common to all agents).

The network communication structure is represented by a directed (or undirected) graph $G = (N, E)$, with the vertex set N and the edge set E. The network is used as a medium to diffuse

the information from an agent to every other agent through local agent interactions over time (▶ Graphs for Modeling Networked Interactions). To accommodate the information spread through the entire network, it is typically assumed that the network communication graph $G = (N, E)$ is strongly connected (▶ Dynamic Graphs, Connectivity of). In the graph, a link (i, j) means that agent $i \in N$ receives the relevant information from agent $j \in N$.

Distributed Algorithms

The algorithms for solving problem (1) are constructed by using standard optimization techniques in combination with a mechanism for information diffusion through local agent interactions. A control point of view for the design of distributed algorithms has a nice exposition in Wang and Elia (2011).

One of the existing optimization techniques is the so-called incremental method, where the information is processed along a directed cycle in the graph. In this approach the estimate is passed from an agent to its neighbor (along the cycle), and only one agent updates at a time (Bertsekas 1997; Blatt et al. 2007; Johansson 2008; Johansson et al. 2009; Nedić and Bertsekas 2000, 2001; Nedić et al. 2001; Rabbat and Nowak 2004; Ram et al. 2009; Tseng 1998); for a detailed literature on incremental methods, see the textbooks Bertsekas (1999) and Bertsekas et al. (2003).

More recently, one of the techniques that gained popularity as a mechanism for information diffusion is a consensus protocol, in which the agent diffuses the information through the network through locally weighted averaging of their incoming data (▶ Averaging Algorithms and Consensus). The problem of reaching a consensus on a particular scalar value, or computing exact averages of the initial values of the agents, has gained an unprecedented interest as a central problem inherent to cooperative behavior in networked systems (Blondel et al. 2005; Boyd et al. 2005; Cao et al. 2005, 2008a,b; Jadbabaie et al. 2003; Olfati-Saber and Murray

2004; Olshevsky 2010; Olshevsky and Tsitsiklis 2006, 2009; Touri 2011; Vicsek et al. 1995; Wan and Lemmon 2009).

Using the consensus technique, a class of distributed algorithms has emerged, as a combination of the consensus protocols and the gradient-type methods. The gradient-based approaches are particularly suitable, as they have a small overhead per iteration and are, in general, robust to various sources of errors and uncertainties.

The technique of using the network as a medium to propagate the relevant information for optimization purpose has its origins in the work by Tsitsiklis (1984), Tsitsiklis et al. (1986), and Bertsekas and Tsitsiklis (1997), where the network has been used to decompose the vector x components across different agents, while all agents share the same objective function.

Algorithms Using Weighted Averaging

The technique has recently been employed in Nedić and Ozdaglar (2009) (see also Nedić and Ozdaglar 2007, 2010) to deal with problems of the form (1) when the agents have different objective functions f_i, but their decisions are fully coupled through the common vector variable x. In a series of recent work (to be detailed later), the following distributed algorithm has emerged. Letting $x_i(k) \in X$ be an estimate (of the optimal decision) at agent i and time k, the next iterate is constructed through two updates. The first update is a consensus-like iteration, whereby, upon receiving the estimates $x_j(k)$ from its (in)neighbors j, the agent i aligns its estimate with its neighbors through averaging, formally given by

$$v_i(k) = \sum_{j \in N_i} w_{ij} x_j(k), \qquad (2)$$

where N_i is the neighbor set

$$N_i = \{j \in N | (j, i) \in E\} \cup \{i\}.$$

The neighbor set N_i includes agent i itself, since the agent always has access to its own information. The scalar w_{ij} is a nonnegative weight that

agent i places on the incoming information from neighbor $j \in N_i$. These weights sum to 1, i.e., $\sum_{j \in N_i} w_{ij} = 1$, thus yielding $v_i(k)$ as a (local) convex combination of $x_j(k)$, $j \in N_i$, obtained by agent i.

After computing $v_i(k)$, agent i computes a new iterate $x_i(k + 1)$ by performing a gradient-projection step, aimed at minimizing its own objective f_i, of the following form:

$$x_i(k + 1) = \prod_X \left[v_i(k) - \alpha(k) \nabla f_i(v_i(k)) \right], \tag{3}$$

where $\Pi_X[x]$ is the projection of a point x on the set X (in the Euclidean norm), $\alpha(k) > 0$ is the stepsize at time k, and $\nabla f_i(z)$ is the gradient of f_i at a point z.

When all functions are zero and X is the entire space P^d, the distributed algorithm (2) and (3) reduces to the linear-iteration method:

$$x_i(k + 1) = \sum_{j \in N_i} w_{ij} x_j(k), \tag{4}$$

which is known as *consensus* or *agreement protocol*. This protocol is employed when the agents in the network wish to align their decision vectors $x_i(k)$ to a common vector $\hat{x}$. The alignment is attained asymptotically (as $k \to \infty$).

In the presence of objective function and constraints, the distributed algorithm in (2) and (3) corresponds to "forced alignment" guided by the gradient forces $\sum_{i=1}^n \nabla f_i(x)$. Under appropriate conditions, the alignment is forced to a common vector x^* that minimizes the network objective $\sum_{i=1}^n f_i(x)$ over the set X. This corresponds to the convergence of the iterates $x_i(k)$ to a common solution $x^* \in X$ as $k \to \infty$ for all agents $i \in N$.

The conditions under which the convergence of $\{x_i(k)\}$ to a common solution $x^* \in X$ occurs are a combination of the conditions needed for the consensus protocol to converge and the conditions imposed on the functions f_i and the stepsize $\alpha(k)$ to ensure the convergence of standard gradient-projection methods. For the consensus part, the conditions should guarantee that the pure consensus protocol in (4) converges to the average $\frac{1}{n} \sum_{i=1}^n x_i(0)$ of the initial agent values. This requirement transfers to the condition that

the weights w_{ij} give rise to a doubly stochastic weight matrix W, whose entries are w_{ij} defined by the weights in (4) and augmented by $w_{ij} = 0$ for $j \notin N_i$.

Intuitively, the requirement that the weight matrix W is doubly stochastic ensures that each agent has the same influence on the system behavior, in a long run. More specifically, the doubly stochastic weights ensure that the system as whole minimizes $\sum_{i=1}^n \frac{1}{n} f_i(x)$, where the factor $1/n$ is seen as portion of the influence of agent i. When the weight matrix W is only row stochastic, the consensus protocol converges to a weighted average $\sum_{i=1}^n \pi_i x_i(0)$ of the agent initial values, where π is the left eigenvector of W associated with the eigenvalue 1. In general, the values π_i can be different for different indices i, and the distributed algorithm in (2) and (3) results in minimizing the function $\sum_{i=1}^n \pi_i f_i(x)$ over X, and thus not solving problem (1).

The distributed algorithm (2) and (3) has been proposed and analyzed in Ram et al. (2010a, 2012), where the convergence had been established for diminishing stepsize rule (i.e., $\sum_k \alpha(k) = \infty$ and $\sum_k \alpha^2(k) < \infty$). As seen from the convergence analysis (see, e.g., Nedić and Ozdaglar 2009; Ram et al. 2012), for the convergence of the method, it is critical that the iterate disagreements $\| x_i(k) - x_j(k) \|$ converge linearly in time, for all $i \neq j$. This fast disagreement decay seems to be indispensable for ensuring the stability of the iterative process (2) and (3).

According to the distributed algorithm (2) and (3), at first, each agent aligns its estimate $x_i(k)$ with the estimates $x_j(k)$ that are received from its neighbors and, then, updates based on its local objective f_i which is to be minimized over $x \in X$. Alternatively, the distributed method can be constructed by interchanging the alignment step and the gradient-projection step. Such a method, while having the same asymptotic performance as the method in (2) and (3), exhibits a somewhat slower (transient) convergence behavior due to a larger misalignment resulting from taking the gradient-based updates at first. This alternative has been initially proposed independently in Lopes and Sayed (2006) (where simulation

results have been reported), in Nedić and Ozdaglar (2007, 2009) (where the convergence analysis for a time-varying networks and a constant stepsize is given), and in Nedić et al. (2008) (with the quantization effects) and further investigated in Lopes and Sayed (2008), Nedić et al. (2010), and Cattivelli and Sayed (2010). More recently, it has been considered in Lobel et al. (2011) for state-dependent weights and in Tu and Sayed (2012) where the performance is compared with that of an algorithm of the form (2) and (3) for estimation problems.

Algorithm Extensions: Over the past years, many extensions of the distributed algorithm in (2) and (3) have been developed, including the following:

(a) *Time-varying communication graphs*: The algorithm naturally extends to the case of time-varying connectivity graphs $\{G(k)\}$, with $G(k) = (N, E(k))$ defined over the node set N and time-varying links $E(k)$. In this case, the weights w_{ij} in (2) are replaced with $w_{ij}(k)$ and, similarly, the neighbor set N_i is replaced with the corresponding time-dependent neighbor set $N_i(k)$ specified by the graph $G(k)$. The convergence of the algorithm (2) and (3) with these modifications typically requires some additional assumptions of the network connectivity over time and the assumptions on the entries in the corresponding weight matrix sequence $\{W(k)\}$, where $w_{ij}(k) = 0$ for $j \notin N_i(k)$. These conditions are the same as those that guarantee the convergence of the (row stochastic) matrix sequence $\{W(k)\}$ to a rank-one row-stochastic matrix, such as a connectivity over some fixed period (of a sliding-time window), the nonzero diagonal entries in $W(k)$, and the existence of a uniform lower bound on positive entries in $W(k)$; see, for example, Cao et al. (2008b), Touri (2011), Tsitsiklis (1984), Nedić and Ozdaglar (2010), Moreau (2005), and Ren and Beard (2005).

(b) *Noisy gradients*: The algorithm in (2) and (3) works also when the gradient computations $\nabla f_i(x)$ in update (3) are erroneous with random errors. This corresponds to using a stochastic gradient $\tilde{\nabla} f_i(x)$ instead of

$\nabla f_i(x)$, resulting in the following stochastic gradient-projection step:

$$x_i(k+1) = \prod_X \left[v_i(k) - \alpha(k)\tilde{\nabla} f_i(v_i(k)) \right]$$

instead of (3). The convergence of these methods is established for the cases when the stochastic gradients are consistent estimates of the actual gradient, i.e.,

$$\mathrm{E}\left[\tilde{\nabla} f_i(v_i(k))|v_i(k) \right] = \nabla f_i(v_i(k)).$$

The convergence of these methods typically requires the use of non-summable but square-summable stepsize sequence $\{\alpha(k)\}$ (i.e., $\sum_k \alpha(k) = \infty$ and $\sum_k \alpha^2(k) < \infty$), e.g., Ram et al. (2010a).

(c) *Noisy or unreliable communication links*: Communication medium is not always perfect and, often, the communication links are characterized by some random noise process. In this case, while agent $j \in N_i$ sends its estimate $x_j(k)$ to agent i, the agent does not receive the intended message. Rather, it receives $x_j(k)$ with some random link-dependent noise $\xi_{ij}(k)$, i.e., it receives $x_j(k) + \xi_{ij}(k)$ instead of $x_j(k)$ (see Kar and Moura 2011; Patterson et al. 2009; Touri and Nedić 2009 for the influence of noise and link failure on consensus). In such cases, the distributed optimization algorithm needs to be modified to include a stepsize for noise attenuation and the standard stepsize for the gradient scaling. These stepsizes are coupled through an appropriate relative-growth conditions which ensure that the gradient information is maintained at the right level and, at the same time, link-noise is attenuated appropriately (Srivastava and Nedić 2011; Srivastava et al. 2010). Other imperfections of the communication links can also be modeled and incorporated into the optimization method, such as link failures and quantization effects, which can be built using the existing results for consensus protocol (e.g., Carli et al. 2007; Kar and Moura 2010, 2011; Nedić et al. 2008).

(d) *Asynchronous implementations*: The method in (2) and (3) has simultaneous updates,

evident in all agents exchanging and updating information in synchronous time steps indexed by k. In some communication settings, the synchronization of agents is impractical, and the agents are using their own clocks which are not synchronized but do tick according to a common time interval. Such communications result in random weights $w_{ij}(k)$ and random neighbor set $N_i(k) \subseteq E$ in (2), which are typically independent and identically distributed (over time). Most common models are random gossip and random broadcast. In the gossip model, at any time, two randomly selected agents i and j communicate and update, while the other agents sleep (Boyd et al. 2005; Kashyap et al. 2007; Ram et al. 2010b; Srivastava 2011; Srivastava and Nedić 2011). In the broadcast model, a random agent i wakes up and broadcasts its estimate $x_i(k)$. Its neighbors that receive the estimate update their iterates, while the other agents (including the agent who broadcasted) do not update (Aysal et al. 2008; Nedić 2011).

(e) *Distributed constraints*: One of the more challenging aspects is the extension of the algorithm to the case when the constraint set X in (1) is given as an intersection of closed convex sets X_i, one set per agent. Specifically, the set X in (1) is defined by

$$X = \bigcap_{i=1}^{n} X_i,$$

where the set X_i is known to agent i only. In this case, the algorithm has a slight modification at the update of $x_i(k+1)$ in (3), where the projection is on the local set X_i instead of X, i.e., the update in (3) is replaced with the following update:

$$x_i(k+1) = \prod_{X_i} \big[v_i(k) - \alpha(k) \nabla f_i(v_i(k)) \big].$$
$$(5)$$

The resulting method (2), (5) converges under some additional assumptions on the sets X_i, such as the nonempty interior assumption (i.e., the set $\bigcap_{i=1}^{n} X_i$ has a nonempty interior), a linear-intersection assumption (each

X_i is an intersection of finitely many linear equality and/or inequality constraints), or the Slater condition (Lee and Nedić 2012; Nedić et al. 2010; Srivastava 2011; Srivastava and Nedić 2011; Zhu and Martínez 2012).

In principle, most of the simple first-order methods that solve a centralized problem of the form (1) can also be distributed among the agents (through the use of consensus protocols) to solve distributed problem (1). For example, the Nesterov dual-averaging subgradient method (Nesterov 2005) can be distributed as proposed in Duchi et al. (2012), a distributed Newton-Raphson method has been proposed and studied in Zanella et al. (2011), while a distributed simplex algorithm has been constructed and analyzed in Bürger et al. (2012). An interesting method based on finding a zero of the gradient $\nabla f = \sum_{i=1}^{n} \nabla f_i$, distributedly, has been proposed and analyzed in Lu and Tang (2012). Some other distributed algorithms and their implementations can be found in Johansson et al. (2007), Tsianos et al. (2012a,b), Dominguez-Garcia and Hadjicostis (2011), Tsianos (2013), Gharesifard and Cortés (2012a), Jakovetic et al. (2011a,b), and Zargham et al. (2012).

Summary and Future Directions

The distributed optimization algorithms have been developed mainly using consensus protocols that are based on weighted averaging, also known as linear-iterative methods. The convergence behavior and convergence rate analysis of these methods combines the tools from optimization theory, graph theory, and matrix analysis. The main drawback of these algorithms is that they require (at least theoretically) the use of doubly stochastic weight matrix W (or $W(k)$ in time-varying case) in order to solve problem (1). This requirement can be accommodated by allowing agents to exchange locally some additional information on the weights that they intend to use or their degree knowledge. However, in general, constructing such doubly stochastic weights distributedly on directed graphs is rather a complex problem (Gharesifard and Cortés 2012b).

As an alternative, which seems a promising direction for future research, is the use of so-called push-sum protocol (or sum-ratio algorithm) for consensus problem (Benezit et al. 2010; Kempe et al. 2003). This direction is pioneered in Tsianos et al. (2012b), Tsianos (2013), and Tsianos and Rabbat (2011) for static graphs and recently extended to directed graphs Nedić and Olshevsky (2013) for an unconstrained version of problem (1).

Another promising direction lies in the use of alternating direction method of multipliers (ADMM) in combination with the graph-Laplacian formulation of consensus constraints $N_i x_i = \sum_{j \in N_i} x_j$. A nice exposure to ADMM method is given in Boyd et al. (2010). The first work to address the development of distributed ADMM over a network is Wei and Ozdaglar (2012), where a static network is considered. Its distributed implementation over time-varying graphs will be an important and challenging task.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Dynamic Graphs, Connectivity of
▶ Graphs for Modeling Networked Interactions
▶ Networked Systems

Recommended Reading

In addition to the below cited literature the useful material relevant to consensus and matrix-product convergence theory includes:

Chatterjee S, Seneta E (1977) Towards consensus: some convergence theorems on repeated averaging. J Appl Probab 14(1):89–97
Cogburn R (1986) On products of random stochastic matrices. Random matrices and their applications. American Mathematical Society, vol. 50, pp 199–213
DeGroot MH (1974) Reaching a consensus. J Am Stat Assoc 69(345):118–121
Lorenz J (2005) A stabilization theorem for continuous opinion dynamics. Physica A: Stat Mech Appl 355:217–223

Rosenblatt M (1965) Products of independent identically distributed stochastic matrices. J Math Anal Appl 11(1):1–10
Shen J (2000) A geometric approach to ergodic non-homogeneous Markov chains. In: T.-X. He (Ed.), Proc. Wavelet Analysis and Multiresolution Methods. Marcel Dekker Inc., New York, vol 212, pp. 341–366
Tahbaz-Salehi A, Jadbabaie A (2010) Consensus over ergodic stationary graph processes. IEEE Trans Autom Control 55(1):225–230
Touri B (2012) Product of random stochastic matrices and distributed averaging. Springer Theses. Springer, Berlin/New York
Wolfowitz J (1963) Products of indecomposable, aperiodic, stochastic matrices. Proc Am Math Soc 14(4):733–737

Bibliography

Aysal T, Yildiz M, Sarwate A, Scaglione A (2008) Broadcast gossip algorithms: design and analysis for consensus. In: Proceedings of the 47th IEEE conference on decision and control, Cancún, pp 4843–4848
Benezit F, Blondel V, Thiran P, Tsitsiklis J, Vetterli M (2010) Weighted gossip: distributed averaging using non-doubly stochastic matrices. In: Proceedings of the 2010 IEEE international symposium on information theory, Austin
Bertsekas D (1997) A new class of incremental gradient methods for least squares problems. SIAM J Optim 7:913–926
Bertsekas D (1999) Nonlinear programming. Athena Scientific, Belmont
Bertsekas D, Tsitsiklis J (1997) Parallel and distributed computation: numerical methods. Athena Scientific, Belmont
Bertsekas D, Nedić A, Ozdaglar A (2003) Convex analysis and optimization. Athena Scientific, Belmont
Blatt D, Hero A, Gauchman H (2007) A convergent incremental gradient algorithm with a constant stepsize. SIAM J Optim 18:29–51
Blondel V, Hendrickx J, Olshevsky A, Tsitsiklis J (2005) Convergence in multiagent coordination, consensus, and flocking. In: Proceedings of IEEE CDC, Seville, pp 2996–3000
Boyd S, Ghosh A, Prabhakar B, Shah D (2005) Gossip algorithms: design, analysis, and applications.

In: Proceedings of IEEE INFOCOM, Miami, vol 3, pp 1653–1664

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1): 1–122

Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Applied mathematics series. Princeton University Press, Princeton

Bürger M, Notarsetfano G, Bullo F, Allgöwer F (2012) A distributed simplex algorithm for degenerate linear programs and multi-agent assignments. Automatica 48(9):2298–2304

Cao M, Spielman D, Morse A (2005) A lower bound on convergence of a distributed network consensus algorithm. In: Proceedings of IEEE CDC, Seville, pp 2356–2361

Cao M, Morse A, Anderson B (2008a) Reaching a consensus in a dynamically changing environment: a graphical approach. SIAM J Control Optim 47(2): 575–600

Cao M, Morse A, Anderson B (2008b) Reaching a consensus in a dynamically changing environment: convergence rates, measurement delays, and asynchronous events. SIAM J Control Optim 47(2):601–623

Carli R, Fagnani F, Frasca P, Taylor T, Zampieri S (2007) Average consensus on networks with transmission noise or quantization. In: Proceedings of European control conference, Kos

Cattivelli F, Sayed A (2010) Diffusion LMS strategies for distributed estimation. IEEE Trans Signal Process 58(3):1035–1048

Dominguez-Garcia A, Hadjicostis C (2011) Distributed strategies for average consensus in directed graphs. In: Proceedings of the IEEE conference on decision and control, Orlando, Dec 2011

Duchi J, Agarwal A, Wainwright M (2012) Dual averaging for distributed optimization: convergence analysis and network scaling. IEEE Trans Autom Control 57(3):592–606

Gharesifard B, Cortés J (2012a) Distributed continuous-time convex optimization on weight-balanced digraphs. http://arxiv.org/pdf/1204.0304.pdf

Gharesifard B, Cortés J (2012b) Distributed strategies for generating weight-balanced and doubly stochastic digraphs. Eur J Control 18(6):539–557

Hendrickx J (2008) Graphs and networks for the analysis of autonomous agent systems. Ph.D. dissertation, Université Catholique de Louvain

Jadbabaie A, Lin J, Morse S (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Autom Control 48(6): 988–1001

Jakovetic D, Xavier J, Moura J (2011a) Cooperative convex optimization in networked systems: augmented lagrangian algorithms with directed gossip communication. IEEE Trans Signal Process 59(8): 3889–3902

Jakovetic D, Xavier J, Moura J (2011b) Fast distributed gradient methods. Available at: http://arxiv.org/abs/1112.2972

Johansson B (2008) On distributed optimization in networked systems. Ph.D. dissertation, Royal Institute of Technology, Stockholm

Johansson B, Rabi M, Johansson M (2007) A simple peer-to-peer algorithm for distributed optimization in sensor networks. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, Dec 2007, pp 4705–4710

Johansson B, Rabi M, Johansson M (2009) A randomized incremental subgradient method for distributed optimization in networked systems. SIAM J Control Optim 20(3):1157–1170

Kar S, Moura J (2010) Distributed consensus algorithms in sensor networks: quantized data and random link failures. IEEE Trans Signal Process 58(3): 1383–1400

Kar S, Moura J (2011) Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs. IEEE J Sel Top Signal Process 5(4):674–690

Kashyap A, Basar T, Srikant R (2007) Quantized consensus. Automatica 43(7):1192–1203

Kempe D, Dobra A, Gehrke J (2003) Gossip-based computation of aggregate information. In: Proceedings of the 44th annual IEEE symposium on foundations of computer science, Cambridge, Oct 2003, pp 482–491

Lee S, Nedić A (2012) Distributed random projection algorithm for convex optimization. IEEE J Sel Top Signal Process 48(6):988–1001 (accepted to appear)

Lobel I, Ozdaglar A, Feijer D (2011) Distributed multi-agent optimization with state-dependent communication. Math Program 129(2):255–284

Lopes C, Sayed A (2006) Distributed processing over adaptive networks. In: Adaptive sensor array processing workshop, MIT Lincoln Laboratory, Lexington, pp 1–5

Lopes C, Sayed A (2008) Diffusion least-mean squares over adaptive networks: formulation and performance analysis. IEEE Trans Signal Process 56(7): 3122–3136

Lu J, Tang C (2012) Zero-gradient-sum algorithms for distributed convex optimization: the continuous-time case. IEEE Trans Autom Control 57(9):2348–2354

Martinoli A, Mondada F, Mermoud G, Correll N, Egerstedt M, Hsieh A, Parker L, Stoy K (2013) Distributed autonomous robotic systems. Springer tracts in advanced robotics. Springer, Heidelberg/New York

Mesbahi M, Egerstedt M (2010) Graph theoretic methods for multiagent networks. Princeton University Press, Princeton

Moreau L (2005) Stability of multiagent systems with time-dependent communication links. IEEE Trans Autom Control 50(2):169–182

Nedić A (2011) Asynchronous broadcast-based convex optimization over a network. IEEE Trans Autom Control 56(6):1337–1351

Nedić A, Bertsekas D (2000) Convergence rate of incremental subgradient algorithms. In: Uryasev S, Pardalos P (eds) Stochastic optimization: algorithms and applications. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 263–304

D

Nedić A, Bertsekas D (2001) Incremental subgradient methods for nondifferentiable optimization. SIAM J Optim 56(1):109–138

Nedić A, Olshevsky A (2013) Distributed optimization over time-varying directed graphs. Available at http://arxiv.org/abs/1303.2289

Nedić A, Ozdaglar A (2007) On the rate of convergence of distributed subgradient methods for multi-agent optimization. In: Proceedings of IEEE CDC, New Orleans, pp 4711–4716

Nedić A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Control 54(1):48–61

Nedić A, Ozdaglar A (2010) Cooperative distributed multi-agent optimization. In: Eldar Y, Palomar D (eds) Convex optimization in signal processing and communications. Cambridge University Press, Cambridge/New York, pp 340–386

Nedić A, Bertsekas D, Borkar V (2001) Distributed asynchronous incremental subgradient methods. In: Butnariu D, Censor Y, Reich S (eds) Inherently parallel algorithms in feasibility and optimization and their applications. Studies in computational mathematics. Elsevier, Amsterdam/New York

Nedić A, Olshevsky A, Ozdaglar A, Tsitsiklis J (2008) Distributed subgradient methods and quantization effects. In: Proceedings of 47th IEEE conference on decision and control, Cancún, Dec 2008, pp 4177–4184

Nedić A, Ozdaglar A, Parrilo PA (2010) Constrained consensus and optimization in multi-agent networks. IEEE Trans Autom Control 55(4):922–938

Nesterov Y (2005) Primal-dual subgradient methods for convex problems, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), Technical report 67

Olfati-Saber R, Murray R (2004) Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans Autom Control 49(9):1520–1533

Olshevsky A (2010) Efficient information aggregation for distributed control and signal processing. Ph.D. dissertation, MIT

Olshevsky A, Tsitsiklis J (2006) Convergence rates in distributed consensus averaging. In: Proceedings of IEEE CDC, San Diego, pp 3387–3392

Olshevsky A, Tsitsiklis J (2009) Convergence speed in distributed consensus and averaging. SIAM J Control Optim 48(1):33–55

Patterson S, Bamieh B, Abbadi A (2009) Distributed average consensus with stochastic communication failures. IEEE Trans Signal Process 57:2748–2761

Rabbat M, Nowak R (2004) Distributed optimization in sensor networks. In: Symposium on information processing of sensor networks, Berkeley, pp 20–27

Ram SS, Nedić A, Veeravalli V (2009) Incremental stochastic sub-gradient algorithms for convex optimization. SIAM J Optim 20(2):691–717

Ram SS, Nedić A, Veeravalli VV (2010a) Distributed stochastic sub-gradient projection algorithms for convex optimization. J Optim Theory Appl 147:516–545

Ram S, Nedić A, Veeravalli V (2010b) Asynchronous gossip algorithms for stochastic optimization: constant stepsize analysis. In: Recent advances in optimization and its applications in engineering: the 14th Belgian-French-German conference on optimization (BFG). Springer-Verlag, Berlin Heidelberg, pp 51–60

Ram SS, Nedić A, Veeravalli VV (2012) A new class of distributed optimization algorithms: application to regression of distributed data. Optim Method Softw 27(1):71–88

Ren W, Beard R (2005) Consensus seeking in multi-agent systems under dynamically changing interaction topologies. IEEE Trans Autom Control 50(5):655–661

Srivastava K (2011) Distributed optimization with applications to sensor networks and machine learning. Ph.D. dissertation, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering

Srivastava K, Nedić A (2011) Distributed asynchronous constrained stochastic optimization. IEEE J Sel Top Signal Process 5(4):772–790

Srivastava K, Nedić A, Stipanović D (2010) Distributed constrained optimization over noisy networks. In: Proceedings of the 49th IEEE conference on decision and control (CDC), Atlanta, pp 1945–1950

Tseng P (1998) An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. SIAM J Optim 8:506–531

Tsianos K (2013) The role of the network in distributed optimization algorithms: convergence rates, scalability, communication/computation tradeoffs and communication delays. Ph.D. dissertation, McGill University, Department of Electrical and Computer Engineering

Tsianos K, Rabbat M (2011) Distributed consensus and optimization under communication delays. In: Proceedings of Allerton conference on communication, control, and computing, Monticello, pp 974–982

Tsianos K, Lawlor S, Rabbat M (2012a) Consensus-based distributed optimization: practical issues and applications in large-scale machine learning. In: Proceedings of the 50th Allerton conference on communication, control, and computing, Monticello

Tsianos K, Lawlor S, Rabbat M (2012b) Push-sum distributed dual averaging for convex optimization. In: Proceedings of the IEEE conference on decision and control, Maui

Tsitsiklis J (1984) Problems in decentralized decision making and computation. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

Tsitsiklis J, Bertsekas D, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE Trans Autom Control 31(9):803–812

Touri B (2011) Product of random stochastic matrices and distributed averaging. Ph.D. dissertation, University of Illinois at Urbana-Champaign, Industrial and Enterprise Systems Engineering

Touri B, Nedić A (2009) Distributed consensus over network with noisy links. In: Proceedings of the 12th international conference on information fusion, Seattle, pp 146–154

Tu S-Y, Sayed A (2012) Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. http://arxiv.org/abs/1205.3993

Vicsek T, Czirok A, Ben-Jacob E, Cohen I, Schochet O (1995) Novel type of phase transitions in a system of self-driven particles. Phys Rev Lett 75(6):1226–1229

Wan P, Lemmon M (2009) Event-triggered distributed optimization in sensor networks. In: Symposium on information processing of sensor networks, San Francisco, pp 49–60

Wang J, Elia N (2011) A control perspective for centralized and distributed convex optimization. In: IEEE conference on decision and control, Florida, pp 3800–3805

Wei E, Ozdaglar A (2012) Distributed alternating direction method of multipliers. In: Proceedings of the 51st IEEE conference on decision and control and European control conference, Maui

Zanella F, Varagnolo D, Cenedese A, Pillonetto G, Schenato L (2011) Newton-Raphson consensus for distributed convex optimization. In: IEEE conference on decision and control, Florida, pp 5917–5922

Zargham M, Ribeiro A, Ozdaglar A, Jadbabaie A (2014) Accelerated dual descent for network flow optimization. IEEE Trans Autom Control 59(4):905–920

Zhu M, Martínez S (2012) On distributed convex optimization under inequality and equality constraints. IEEE Trans Autom Control 57(1):151–164

DOC

▶ Discrete Optimal Control

Dynamic Graphs, Connectivity of

Michael M. Zavlanos [1] and George J. Pappas[2]
[1]Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC, USA
[2]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA

Abstract

Dynamic networks have recently emerged as an efficient way to model various forms of interaction within teams of mobile agents, such as

sensing and communication. This article focuses on the use of graphs as models of wireless communications. In this context, graphs have been used widely in the study of robotic and sensor networks and have provided an invaluable modeling framework to address a number of coordinated tasks ranging from exploration, surveillance, and reconnaissance to cooperative construction and manipulation. In fact, the success of these stories has almost always relied on efficient information exchange and coordination between the members of the team, as seen, e.g., in the case of distributed state agreement where multi-hop communication has been proven necessary for convergence and performance guarantees.

Keywords

Algebraic graph theory; Convex optimization; Distributed and hybrid control; Graph connectivity

Introduction

Communication in networked dynamical systems has typically relied on constructs from graph theory, with disc-based and weighted-proximity graphs gaining the most popularity; see Fig. 1a, b. Besides their simplicity, these models owe their popularity to their resemblance to radio signal strength models, where the signals attenuate with the distance (Neskovic et al. 2000; Pahlavan and Levesque 1995; Parsons 2000). In this context, multi-hop communication becomes equivalent to network connectivity, defined as the property of a graph to transmit information between any pair of its nodes; see Fig. 1c.

Specifically, let $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}(t), \mathcal{W}(t)\}$ denote a graph on n nodes that can be robots or mobile sensors, so that $\mathcal{V} = \{1, \ldots, n\}$ is the set of vertices, $\mathcal{E}(t) \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges at time t, and $\mathcal{W}(t) = \{w_{ij}(t) \mid (i, j) \in \mathcal{V} \times \mathcal{V}\}$ is a set of weights so that $w_{ij}(t) = 0$ if $(i, j) \notin \mathcal{E}(t)$ and $w_{ij}(t) > 0$ otherwise. If $w_{ij}(t) = w_{ji}(t)$ for all pairs of nodes i, j, then the graph is called
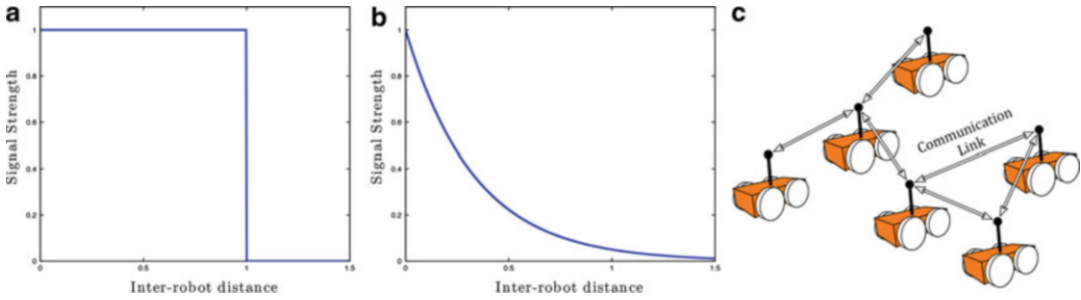
Dynamic Graphs, Connectivity of, Fig. 1 (**a**) Disc-based model of communication; (**b**) Weighted, proximity-based model of communication; (**c**) Connected network of mobile robots

undirected; otherwise it is called directed. The weights in $\mathcal{W}(t)$ typically model signal strength or channel reliability, as per the disc-based and weighted-proximity models in Fig. 1a, b. In these models communication between nodes is related to their pairwise distance, giving rise to the dynamic or time-varying nature of the graph $\mathcal{G}(t)$ due to node mobility. Given an undirected dynamic graph $\mathcal{G}(t)$, we say that this graph is *connected* at time t if there exists a path, i.e., a sequence of distinct vertices such that consecutive vertices are adjacent, between any two vertices in $\mathcal{G}(t)$. In the case of directed graphs, two notions of connectivity are defined. A directed graph $\mathcal{G}(t)$ is called *strongly connected* if there exists a directed path between any two of its vertices or equivalently, if every vertex is reachable from any other vertex. On the other hand, a directed graph is called *weakly connected* if replacing all directed edges by undirected edges produces a connected undirected graph. Finally, a collection of graphs $\{\mathcal{G}(t) \mid t = t_0, \ldots, t_k\}$ is called *jointly connected* over time if the union graph $\cup_{t=t_0}^{t_k} \mathcal{G}(t) = \{\mathcal{V}, \cup_{t=t_0}^{t_k} \mathcal{E}(t)\}$ is connected. Clearly checking for the existence of paths between all pairs of nodes in a graph is difficult, especially so as the number of nodes in the graph increases. For this reason, equivalent, algebraic representations of graphs are employed that allow for efficient algebraic ways to check for connectivity, as we discuss in the following section.

While connectivity is necessary for information propagation in a network, it is also relevant to the performance of many networked dynamical processes, such as synchronization and gossiping, via its relation to the network eigenvalue spectra (Preciado 2008). For example, the spectrum of the Laplacian matrix of a network plays a key role in the analysis of synchronization in networks of nonlinear oscillators (Pecora and Carrollg 1998; Preciado and Verghese 2005), distributed algorithms (Lynch 1997), and decentralized control problems (Fax and Murray 2004; Olfati Saber and Murray 2004). Similarly, the spectrum of the adjacency matrix determines the speed of viral information spreading in a network (Van Mieghem et al. 2009). Additionally, more robust versions of connectivity, such as *k-node* or *k-edge* connectivity, can be used to introduce robustness of a network to node or link failures, respectively (Zavlanos and Pappas 2005, 2008).

Graph-Theoretic Connectivity Control

Connectivity Using the Graph Laplacian Matrix

A metric that is typically employed to capture connectivity of dynamic networks is the second smallest eigenvalue $\lambda_2(L)$ of the Laplacian matrix $L \in \mathbb{R}^{n \times n}$ of the graph, also known as the algebraic connectivity or Fiedler value of the graph. For a weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, the entries of the Laplacian matrix are typically related to the weights in $\mathcal{W}$ so that the i, j entry of L is given by $[L]_{ij} = \sum_{j=1}^{n} w_{ij}$ if $i = j$ and $[L]_{ij} = -w_{ij}$ if $i \neq j$. The Laplacian matrix of an undirected graph is always a symmetric, positive semidefinite matrix whose smallest eigenvalue $\lambda_1(L)$ is identically zero

with corresponding eigenvector the vector of all entries equal to one. Additionally, the algebraic connectivity $\lambda_2(L)$ is a concave function of the Laplacian matrix that is positive if and only if the graph is connected (Fiedler 1973; Godsil and Royle 2001; Merris 1994; Mohar 1991).

As the algebraic connectivity $\lambda_2(L)$ plays a critical role in determining whether a graph is connected or not, a number of methods have been proposed for its decentralized estimation and control. These range from methods that employ market-based control to underestimate the algebraic connectivity and accordingly control the network structure (Zavlanos and Pappas 2008) to methods that enforce the states of the nodes to oscillate at frequencies that correspond to the Laplacian eigenvalues and then use fast Fourier transform to estimate these eigenvalues (Franceschelli et al. 2013), to methods that iteratively update the interval where the algebraic connectivity is supposed to lie (Montijano et al. 2011), and to methods that rely on the power iteration method and its variants (DeGennaro and Jadbabaie 2006; Kempe and McSherry 2008; Knorn et al. 2009; Oreshkin et al. 2010; Sabattini et al. 2011; Yang et al. 2010). All the above techniques are often integrated with appropriate controllers to regulate mobility of the nodes while ensuring connectivity of the network. Another way that $\lambda_2(L)$ can be used to ensure connectivity of dynamic graphs is via optimization-based methods that maximize it away from its zero value. Such approaches were initially centralized as connectivity is a global property of a graph (Kim and Mesbahi 2006), although recently distributed subgradient algorithms (DeGennaro and Jadbabaie 2006) as well as non-iterative decomposition techniques (Simonetto et al. 2013) have also been proposed. As the algebraic connectivity is a non-differentiable function of the Laplacian matrix, designing continuous feedback controllers to maintain it positive definite is a challenging task. This problem was overcome in Zavlanos and Pappas (2007) via the use of gradient flows that maintain positive definiteness of the determinant of the projected Laplacian matrix to the space that is perpendicular to eigenvector of ones.

Connectivity Using the Graph Adjacency Matrix

Alternatively, connectivity can be captured by the sum of powers $\sum_{k=0}^{K} A^k$ of the adjacency matrix $A \in \mathbb{R}^{n \times n}$ of the network for $K \leq n - 1$. The entries of the adjacency matrix are typically related to the weights in $\mathcal{W}$ as $[A]_{ij} = w_{ij}$. For disc-based graphs as in Fig. 1a, the i, j entry of the kth power of the adjacency matrix $[A^k]_{ij}$ captures the number of paths of length k between nodes i and j; for weighted graphs, $[A^k]_{ij}$ captures a weighted sum of those paths. Therefore, the entries of $\sum_{k=0}^{K} A^k$ represent the number of paths up to length K between every pair of nodes in the graph (Godsil and Royle 2001). By definition of graph connectivity, if all entries of $\sum_{k=0}^{K} A^k$ are positive for $K = n - 1$, then the network is connected. Clearly, for $K < n - 1$, not all entries of $\sum_{k=0}^{K} A^k$ are necessarily positive, even if the graph is connected. Maintaining positive definiteness of the positive entries of $\sum_{k=0}^{K} A^k$ of an initially connected graph maintains paths of length K between the corresponding nodes and, as shown in Zavlanos and Pappas (2005), is sufficient to maintain connectivity of the graph throughout.

The ability to capture graph connectivity using the adjacency matrix has given rise to optimization-based connectivity controllers (Srivastava and Spong 2008; Zavlanos and Pappas 2005) that are often centralized due to the multi-hop dependencies between nodes due to the powers of the adjacency matrix. Since smaller powers correspond to shorter dependencies (paths), decentralization is possible as K decreases. If $K = 1$, connectivity maintenance reduces to preserving the pairwise links between the nodes in an initially connected network. Since the adjacency matrix of weighted graphs is often a differentiable function, this approach can result in continuous feedback solution techniques. Discrete-time approaches are discussed in Ando et al. (1999), Notarstefano et al. (2006), and Bullo et al. (2009), while Spanos and Murray (2004), Dimarogonas and Kyriakopoulos (2008), Cornejo and Lynch (2008), Yao and Gupta (2009), Zavlanos et al. (2007), and Ji and Egerstedt (2007) rely on local gradients that

may also incorporate switching in the case of link additions. Switching between arbitrary spanning topologies has also been studied in the literature, with the spanning subgraphs being updated by local auctions (Zavlanos and Pappas 2008), distributed spanning tree algorithms (Wagenpfeil et al. 2009), combination of information dissemination algorithms and graph picking games (Schuresko and Cortes 2009b), or intermediate rendezvous (Schuresko and Cortes 2009a; Spanos and Murray 2005). This class of approaches is typically hybrid, combining continuous link maintenance and discrete topology control. The algebraic connectivity $\lambda_2(L)$ and number of paths $\sum_{k=0}^{K} A^k$ metrics can also be combined to give controllers that maintain connectivity, while enforcing desired multi-hop neighborhoods for all agents (Stump et al. 2008).

A recent, comprehensive survey on graph-theoretic approaches for connectivity control of dynamic graphs can be found in Zavlanos et al. (2011).

Applications in Mobile Robot Network Control

Methods to control connectivity of dynamic graphs have been successfully applied to multiple scenarios that require network connectivity to achieve a global coordinated objective. Indicative of the impact of this work is recent literature on connectivity preserving rendezvous (Ando et al. 1999; Cortes et al. 2006; Dimarogonas and Kyriakopoulos 2008; Ganguli et al. 2009; Ji and Egerstedt 2007), flocking (Zavlanos et al. 2007, 2009), and formation control (Ji and Egerstedt 2007; Schuresko and Cortes 2009a), where so far connectivity had been an assumption. Further extensions and contributions involve connectivity control for double integrator agents (Notarstefano et al. 2006), agents with bounded inputs (Ajorlou and Aghdam 2010; Ajorlou et al. 2010; Dimarogonas and Johansson 2008), and indoor navigation (Stump et al. 2008), as well as for communication based on radio signal strength (Hsieh et al. 2008; Mostofi 2009;

Powers and Balch 2004; Wagner and Arkin 2004) and visibility constraints (Anderson et al. 2003; Ando et al. 1999; Arkin and Diaz 2002; Flocchini et al. 2005; Ganguli et al. 2009). Periodic connectivity for robot teams that need to occasionally split in order to achieve individual objectives (Hollinger and Singh 2010; Zavlanos 2010) and sufficient conditions for connectivity in leader-follower networks (Gustavi et al. 2010) also adds to the list. Early experimental results have demonstrated efficiency of these algorithms also in practice (Hollinger and Singh 2010; Michael et al. 2009; Tardioli et al. 2010).

Summary and Future Directions

Although graphs provide a simple abstraction of inter-robot communications, it has long been recognized that since links in a wireless network do not entail tangible connections, associating links with arcs on a graph can be somewhat arbitrary. Indeed, topological definitions of connectivity start by setting target signal strengths to draw the corresponding graph. Even small differences in target strengths might result in dramatic differences in network topology (Lundgren et al. 2002). As a result, graph connectivity is necessary but not nearly sufficient to guarantee communication integrity, interpreted as the ability of a network to support desired communication rates.

To address these challenges, a new body of work is recently appearing that departs from traditional graph-based models of communication. Specifically, Zavlanos et al. (2013) employs a simple, yet effective, modification that relies on weighted graph models with weights that capture the packet error probability of each link (De-Couto et al. 2006). When using reliabilities as link metrics, it is possible to model routing and scheduling problems as optimization problems that accept link reliabilities as inputs (Ribeiro et al. 2007, 2008). The key idea proposed in Zavlanos et al. (2013) is to define connectivity in terms of communication rates and to use optimization formulations to describe optimal operating points of wireless networks. Then, the

communication variables are updated in discrete time via a distributed gradient descent algorithm on the dual function, while robot motion is regulated in continuous time by means of appropriate distributed barrier potentials that maintain desired communication rates. Related approaches consider optimal communications based on T-slot time averages of the primal variables for general mobility schemes Neely (2010), as well as optimization of mobility and communications based on the end-to-end bit error rate between nodes (Ghaffarkhah and Mostofi 2011; Yan and Mostofi 2012).

Cross-References

▶ Flocking in Networked Systems
▶ Graphs for Modeling Networked Interactions

Bibliography

Ajorlou A, Aghdam AG (2010) A class of bounded distributed controllers for connectivity preservation of unicycles. In: Proceedings of the 49th IEEE conference on decision and control, Atlanta, pp 3072–3077

Ajorlou A, Momeni A, Aghdam AG (2010) A class of bounded distributed control strategies for connectivity preservation in multi-agent systems. IEEE Trans Autom Control 55(12):2828–2833

Anderson SO, Simmons R, Goldberg D (2003) Maintaining line-of-sight communications networks between planetray rovers. In: Proceedings of the 2003 IEEE/RSJ international conference on intelligent robots and systems, Las Vegas, pp 2266–2272

Ando H, Oasa Y, Suzuki I, Yamashita M (1999) Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Trans Robot Autom 15(5):818–828

Arkin RC, Diaz J (2002) Line-of-sight constrained exploration for reactive multiagent robotic teams. In: Proceedings of the 7th international workshop on advanced motion control, Maribor, pp 455–461

Bullo F, Cortes J, Martinez S (2009) Distributed control of robotic networks. Applied Mathematics Series. Princeton University Press, Princeton

Cornejo A, Lynch N (2008) Connectivity service for mobile ad-hoc networks. In: Proceedings of the 2nd IEEE international conference on self-adaptive and self-organizing systems workshops, pp 292–297

Cortes J, Martinez S, Bullo F (2006) Robust rendezvous for mobile autonomous agents via proximity graphs

in arbitrary dimensions. IEEE Trans Autom Control 51(8):1289–1298

DeCouto D, Aguayo D, Bicket J, Morris R (2006) A high-throughput path metric for multihop wireless routing. In: Proceedings of the international ACM conference on mobile computing and networking, San Diego, pp 134–146

DeGennaro MC, Jadbabaie A (2006) Decentralized control of connectivity for multi-agent systems. In: Proceedings of the 45th IEEE conference on decision and control, San Diego, pp 3628–3633

Dimarogonas DV, Johansson KH (2008) Decentralized connectivity maintenance in mobile networks with bounded inputs. In Proceedings of the IEEE international conference on robotics and automation, Pasadena, pp 1507–1512

Dimarogonas DV, Kyriakopoulos KJ (2008) Connectedness preserving distributed swarm aggregation for multiple kinematic robots. IEEE Trans Robot 24(5):1213–1223

Fax A, Murray RM (2004) Information flow and cooperative control of vehicle formations. IEEE Trans Autom Control 49:1465–1476

Fiedler M (1973) Algebraic connectivity of graphs. Czechoslovak Math J 23(98):298–305

Flocchini P, Prencipe G, Santoro N, Widmayer P (2005) Gathering of asynchronous oblivious robots with limited visibility. Theor Comput Sci 337(1–3):147–168

Franceschelli M, Gasparri A, Giua A, Seatzu C (2013) Decentralized estimation of laplacian eigenvalues in multi-agent systems. Automatica 49(4):1031–1036

Ganguli A, Cortes J, Bullo F (2009) Multirobot rendezvous with visibility sensors in nonconvex environments. IEEE Trans Robot 25(2):340–352

Ghaffarkhah A, Mostofi Y (2011) Communication-aware motion planning in mobile networks. IEEE Trans Autom Control Spec Issue Wirel Sens Actuator Netw 56(10):2478–248

Godsil C, Royle G (2001) Algebraic graph theory, Graduate Texts in Mathematics, vol 207. Springer, Berlin

Gustavi T, Dimarogonas DV, Egerstedt M, Hu X (2010) Sufficient conditions for connectivity maintenance and rendezvous in leader-follower networks. Automatica 46(1):133–139

Hollinger G, Singh S (2010) Multi-robot coordination with periodic connectivity. In: Proceedings of the IEEE international conference on robotics and automation, Anchorage, Alaska, pp 4457–4462

Hsieh MA, Cowley A, Kumar V, Taylor C (2008) Maintaining network connectivity and performance in robot teams. J Field Robot 25(1–2):111–131

Ji M, Egerstedt M (2007) Coordination control of multi-agent systems while preserving connectedness. IEEE Trans Robot 23(4):693–703

Kempe D, McSherry F (2008) A decentralized algorithm for spectral analysis. J Comput Syst Sci 74(1):70–83

Kim Y, Mesbahi M (2006) On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. IEEE Trans Autom Control 51(1):116–120

Knorn F, Stanojevic R, Corless M, Shorten R (2009) A framework for decentralized feedback connectivity control with application to sensor networks. Int J Control 82(11):2095–2114

Lundgren H, Nordstrom E, Tschudin C (2002) The gray zone problem in ieee 802.11b based ad hoc networks. ACM SIGMOBILE Mobile Comput Commun Rev 6(3):104–105

Lynch N (1997) Distributed algorithms. Morgan Kaufmann, San Francisco

Merris R (1994) Laplacian matrices of a graph: a survey. Linear Algebra Appl 197:143–176

Michael N, Zavlanos MM, Kumar V, Pappas GJ (2009) Maintaining connectivity in mobile robot networks. In: Experimental robotics. Tracts in advanced robotics. Springer, Berlin/Heidelberg, pp 117–126

Mohar B (1991) The laplacian spectrum of graphs. In: Alavi Y, Chartrand G, Ollermann O, Schwenk A (Eds) Graph theory, combinatorics, and applications. Wiley, New York, pp 871–898

Montijano E, Montijano JI, Sagues C (2011) Adaptive consensus and algebraic connectivity estimation in sensor networks with chebyshev polynomials. In: Proceedings of the 50th IEEE conference on decision and control, Orlando, pp 4296–4301

Mostofi Y (2009) Decentralized communication-aware motion planning in mobile networks: an information-gain approach. J Intell Robot Syst 56(1–2):233–256

Neely MJ (2010) Universal scheduling for networks with arbitrary traffic, channels, and mobility. In: Proceedings of the 49th IEEE conference on decision and control, Altanta, pp 1822–1829

Neskovic A, Neskovic N, Paunovic G (2000) Modern approaches in modeling of mobile radio systems propagation environment. IEEE Commun Surv 3(3):1–12

Notarstefano G, Savla K, Bullo F, Jadbabaie A (2006) Maintaining limited-range connectivity among second-order agents. In: Proceedings of the 2006 American control conference, Minneapolis, pp 2124–2129

Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans Autom Control 49:1520–1533

Oreshkin BN, Coates MJ, Rabbat MG (2010) Optimization and analysis of distributed averaging with short node memory. IEEE Trans Signal Process 58(5):2850–2865

Pahlavan K, Levesque AH (1995) Wireless information networks. Willey, New York

Parsons JD (2000) The mobile radio propagation channel. Willey, Chichester

Pecora L, Carrollg T (1998) Master stability functions for synchronized coupled systems. Phys Rev Lett 80:2109–2112

Powers M, Balch T (2004) Value-based communication preservation for mobile robots. In: Proceedings of the 7th international symposium on distributed autonomous robotic systems, Toulouse

Preciado V (2008) Spectral analysis for stochastic models of large-scale complex dynamical networks. Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT

Preciado V, Verghese G (2005) Synchronization in generalized erdös-rényi networks of nonlinear oscillators. In: 44th IEEE conference on decision and control, Seville, Spain, pp 4628–463

Ribeiro A, Luo Z-Q, Sidiropoulos ND, Giannakis GB (2007) Modelling and optimization of stochastic routing for wireless multihop networks. In: Proceedings of the 26th annual joint conference of the IEEE Computer and Communications Societies (INFOCOM), Anchorage, pp 1748–1756

Ribeiro A, Sidiropoulos ND, Giannakis GB (2008) Optimal distributed stochastic routing algorithms for wireless multihop networks. IEEE Trans Wirel Commun 7(11):4261–4272

Sabattini L, Chopra N, Secchi C (2011) On decentralized connectivity maintenance for mobile robotic systems. In: Proceedings of the 50th IEEE conference on decision and control, Orlando, pp 988–993

Schuresko M, Cortes J (2009a) Distributed tree rearrangements for reachability and robust connectivity. In: Hybrid systems: computetation and control. Lecture notes in computer science, vol 5469. Springer, Berlin/New York, pp 470–474

Schuresko M, Cortes J (2009b) Distributed motion constraints for algebraic connectivity of robotic networks. J Intell Robot Syst 56(1–2):99–126

Simonetto A, Kaviczky T, Babuska R (2013) Constrained distributed algebraic connectivity maximization in robotic networks. Automatica 49(5):1348–1357

Spanos DP, Murray RM (2004) Robust connectivity of networked vehicles. In: Proceedings of the 43rd IEEE conference on decision and control, Bahamas, pp 2893–2898

Spanos DP, Murray RM (2005) Motion planning with wireless network constraints. In: Proceedings of the 2005 American control conference, Portland, pp 87–92

Srivastava K, Spong MW (2008) Multi-agent coordination under connectivity constraints. In: Proceedings of the 2008 American control conference, Seattle, pp 2648–2653

Stump E, Jadbabaie A, Kumar V (2008) Connectivity management in mobile robot teams. In: Proceedings of the IEEE international conference on robotics and automation, Pasadena, pp 1525–1530

Tardioli D, Mosteo AR, Riazuelo L, Villarroel JL, Montano L (2010) Enforcing network connectivity in robot team missions. Int J Robot Res 29(4):460–480

Van Mieghem P, Omic J, Kooij R (2009) Virus spread in networks. IEEE/ACM Trans Networking 17(1):1–14

Wagenpfeil J, Trachte A, Hatanaka T, Fujita M, Sawodny O (2009) A distributed minimum restrictive connectivity maintenance algorithm. In: Proceedings of the 9th international symposium on robot control, Gifu

Wagner AR, Arkin RC (2004) Communication-sensitive multi-robot reconnaissance. In: Proceedings of the

IEEE international conference on robotics and automation, New Orleans, pp 2480–2487

Yan Y, Mostofi Y (2012) Robotic router formation in realistic communication environments. IEEE Trans Robot 28(4):810–827

Yang P, Freeman RA, Gordon GJ, Lynch KM, Srinivasa SS, Sukthankar R (2010) Decentralized estimation and control of graph connectivity for mobile sensor networks. Automatica 46(2): 390–396

Yao Z, Gupta K (2009) Backbone-based connectivity control for mobile networks. In: Proceedings IEEE international conference on robotics and automation, Kobe, pp 1133–1139

Zavlanos MM (2010) Synchronous rendezvous of very-low-range wireless agents. In: Proceedings of the 49th IEEE conference on decision and control, Atlanta, pp 4740–4745

Zavlanos MM, Pappas GJ (2005) Controlling connectivity of dynamic graphs. In: Proceedings of the 44th IEEE conference on decision and control and European control conference, Seville, pp 6388–6393

Zavlanos MM, Pappas GJ (2007) Potential fields for maintaining connectivity of mobile networks. IEEE Trans Robot 23(4):812–816

Zavlanos MM, Pappas GJ (2008) Distributed connectivity control of mobile networks. IEEE Trans Robot 24(6):1416–1428

Zavlanos MM, Jadbabaie A, Pappas GJ (2007) Flocking while preserving network connectivity. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, pp 2919–2924

Zavlanos MM, Tanner HG, Jadbabaie A, Pappas GJ (2009) Hybrid control for connectivity preserving flocking. IEEE Trans Autom Control 54(12):2869–2875

Zavlanos MM, Egerstedt MB, Pappas GJ (2011) Graph theoretic connectivity control of mobile robot networks. Proc IEEE Spec Issue Swarming Nat Eng Syst 99(9):1525–154

Zavlanos MM, Ribeiro A, Pappas GJ (2013) Network integrity in mobile robotic networks. IEEE Trans Autom Control 58(1):3–18

Dynamic Noncooperative Games

David A. Castañón
Boston University, Boston, MA, USA

Abstract

In this entry, we present models of dynamic noncooperative games, solution concepts and algorithms for finding game solutions. For the sake of exposition, we focus mostly on finite games, where the number of actions available to each player is finite, and discuss briefly extensions to infinite games.

Keywords

Extensive form games; Finite games; Nash equilibrium

Introduction

Dynamic noncooperative games allow multiple actions by individual players, and include explicit representations of the information available to each player for selecting its decision. Such games have a complex temporal order of play and an information structure that reflects uncertainty as to what individual players know when they have to make decisions. This temporal order and information structure is not evident when the game is represented as a static game between players that select strategies. Dynamic games often incorporate explicit uncertainty in outcomes, by representing such outcomes as actions taken by a random player (called chance or "Nature") with known probability distributions for selecting its actions.

We focus our exposition on models of finite games and discuss briefly extensions to infinite games at the end of the entry.

Finite Games in Extensive Form

The *extensive form* of a game was introduced by von Neumann (1928) and later refined by Kuhn (1953) to represent explicitly the order of play, the information and actions available to each player for making decisions at each of their turns, and the payoffs that players receive after a complete set of actions. Let $I = \{0, 1, \ldots, n\}$ denote the set of players in a game, where player 0 corresponds to Nature. The extensive form is represented in terms of a game tree, consisting of a rooted tree with nodes $\mathcal{N}$ and edges $\mathcal{E}$. The root node represents the initial state of the game. Nodes x in this tree correspond to positions or "states" of the game. Any non-root node with

more than one incident edge is an internal node of the tree and is a *decision* node associated with a player in the game; the root node is also a decision node. Each decision node x has a player $o(x) \in I$ assigned to select a decision from a finite set of admissible decisions $A(x)$. Using distance from the root node to indicate direction of play, each edge that follows node x corresponds to an action in $A(x)$ taken by player $p(x)$, which evolves the state of the game into a subsequent node x'.

Non-root nodes with only one incident edge are terminal nodes, which indicate the end of the game; such nodes represent outcomes of the game. The unique path from the root node to a terminal node is a called a *play* of the game. For each outcome node x, there are payoff functions $J_i(x)$ associated with each player $i \in \{1, \ldots, n\}$.

The above form represents the different players, the order in which players select actions and their possible actions, and the resulting payoffs to each player from a complete set of plays of the game. The last component of interest is to represent the information available for players to select decisions at each decision node. Due to the tree structure of the extensive form of a game, each decision node x contains exact information on all of the previous actions taken that led to state x. In games of *perfect information*, each player knows exactly the decision node x at which he/she is selecting an action. To represent imperfect information, the extensive form uses the notion of an *information set*, which represents a group of decision nodes, associated with the same player, where the information available to that player is that the game is in one of the states in that information set. Formally, let $\mathcal{N}_i \subset \mathcal{N}$ be the set of decision nodes associated with player i, for $i \in I$. Let $\mathcal{H}_i$ denote a partition of $\mathcal{N}_i$ so that, for each set $h_i^k \in \mathcal{H}_i$, we have the following properties:

- If $x, x' \in h_i^k$, then they have the same admissible actions: $A(x) = A(x')$.
- If $x, x' \in h_i^k$, then they cannot both belong to a play of the game; that is, both x and x' cannot be on a path from the root node to an outcome.

Elements h_i^k for some player i are the *information sets*. Each decision node x belongs to one and
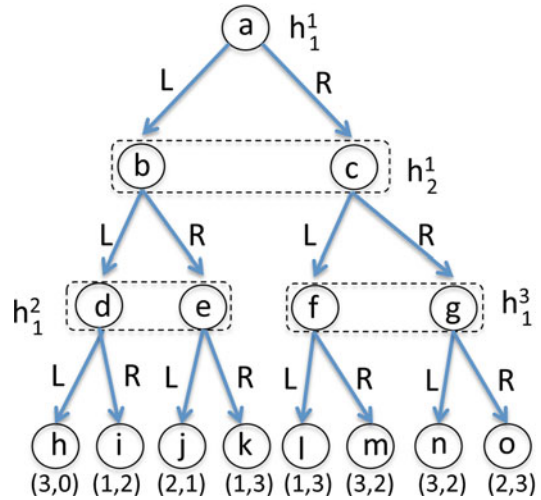
Dynamic Noncooperative Games, Fig. 1 Illustration of game in extensive form

only one information set associated with player $p(x)$. The constraints above ensure that, for each information set, there is a unique player identified to select actions, and the set of admissible actions is unambiguously defined. Denote by $A(h_i^k)$ the set of admissible actions at information set h_i^k. The last condition is a causality condition that ensures that a player who has selected a previous decision remembers that he/she has already made that previous decision.

Figure 1 illustrates an extensive form for a two-person game. Player 1 has two actions in any play of the game, whereas player 2 has only 1 action. Player 1 starts the game at the root node a; the information set h_2^1 shows that player 2 is unaware of this action, as both nodes that descend from node a are in this information set. After player 2's action, player 1 gets to select a second action. However, the information sets h_1^2, h_1^3 indicate that player 1 recalls what earlier action he/she selected, but he/she has not observed the action of player 2. The terminal nodes indicate the payoffs to player 1 and player 2 as an ordered pair.

Strategies and Equilibrium Solutions

A pure strategy γ_i for player $i \in \{1, \ldots, n\}$ is a function that maps each information set of player i into an admissible decision. That is,

$$\gamma_i : \mathcal{H}_i \to A$$

such that $\gamma_i(h_i^k) \in A(h_i^k)$. The set of pure strategies for player i is denoted as Γ_i.

Note that pure strategies do not include actions selected by Nature. Nature's actions are selected using probability distributions over the choices available for the information sets corresponding to Nature, where the choice at each information set is made independently of other choices. For finite games, the number of pure strategies for each player is also finite. Given a tuple of pure strategies $\underline{\gamma} = (\gamma_1, \ldots, \gamma_n)$, we can define the probability of the play of the game resulting in outcome node x as $\pi(x)$, computed as follows: Each outcome node has a unique path (the play) p_{rx} from root node r. We initialize $\pi(r) = 1$ and node $n = r$. If the player at node n is Nature and the next node in p_{rx} is n', then $\pi(n') = \pi(n) * p(n, n')$, where $p(n, n')$ is the probability that Nature chooses the action that leads to n'. Otherwise, let $i = p(n)$ be the player at node n and let $h_i^k(n)$ denote the information set containing node n. Then, if $\gamma^i(h_i^k(n)) = a(n, n')$, let $\pi(n') = \pi(n)$, where $a(n, n')$ is the action in $A(n)$ that leads to n'; otherwise, set $\pi(n') = 0$. The above process is repeated letting $n' = n$, until n' equals the terminal node x. Using these probabilities, the resulting expected payoff to player i is

$$\mathcal{J}_i(\underline{\gamma}) = \sum_{x \text{ terminal}, \, x \in \mathcal{N}} \pi(x) J_i(x)$$

This representation of payoffs in terms of strategies transforms the game from an extensive form representation to a strategic or normal form representation, where the concept of dynamics and information has been abstracted away. The resulting strategic form looks like a static game as discussed in the encyclopedia entry ▸ Strategic Form Games and Nash Equilibrium, where each player selects his/her strategy from a finite set, resulting in a vector of payoffs for the players. Using these payoffs, one can now define solution concepts for the game. Let the notation $\underline{\gamma}_{-i} = (\gamma_1, \ldots, \gamma_{i-1}, \gamma_{i+1}, \gamma_n)$ denote the set of strategies in a tuple excluding the ith strategy.

A Nash equilibrium solution is a tuple of feasible strategies $\underline{\gamma}^* = (\gamma_1^*, \ldots, \gamma_n^*)$ such that

$$\mathcal{J}_i(\underline{\gamma}^*) \geq \mathcal{J}_i(\underline{\gamma}_{-i}^*, \gamma_i) \text{ for all } \gamma_i \in \Gamma_i,$$
$$\text{for all } i \in \{1, \ldots, n\} \qquad (1)$$

The special case of two-person games where $\mathcal{J}_1(\underline{\gamma}) = -\mathcal{J}_2(\underline{\gamma})$ are known as zero-sum games.

As discussed in the encyclopedia entry on static games (▸ Strategic Form Games and Nash Equilibrium), the existence of Nash equilibria or even saddle point strategies in terms of pure strategies is not guaranteed for finite games. Thus, one must consider the use of *mixed strategies*. A mixed strategy μ_i for player $i \in \{1, \ldots, n\}$ is a probability distribution over the set of pure strategies Γ_i. The definition of payoffs can be extended to mixed strategies by averaging the payoffs associated with the pure strategies, as

$$\mathcal{J}_i(\underline{\mu}_i) = \sum_{\gamma_1 \in \Gamma_1} \cdots \sum_{\gamma_n \in \Gamma_n} \mu_1(\gamma_1) \cdots \mu_n(\gamma_n)$$
$$\mathcal{J}_i(\gamma_1, \ldots, \gamma_n)$$

Denote the set of probability distributions over Γ_i as $\Delta(\Gamma_i)$. An n-tuple $\underline{\mu}^* = (\mu_1, \ldots, \mu_n)$ of mixed strategies is said to be a Nash equilibrium if

$$\mathcal{J}_i(\underline{\mu}^*) \geq \mathcal{J}_i(\underline{\mu}_{-i}^*, \mu_i) \text{ for all } \mu_i$$
$$\in \Delta(\Gamma_i), \text{ for all } i \in \{1, \ldots, n\}$$

Theorem 1 (Nash 1950, 1951) *Every finite n-person game has at least one Nash equilibrium point in mixed strategies.*

Mixed strategies suggest that each player's randomization occurs before the game is played, by choosing a strategy at random from its choices of pure strategies. For games in extensive form, one can introduce a different class of strategies where a player makes a random choice of action at each information set, according to a probability distribution that depends on the specific information set. The choice of action is selected independently at each information set according

to a selected probability distribution, in a manner similar to how Nature's actions are selected. These random choice strategies are known as *behavior strategies*. Let $\Delta(A(h_i^k))$ denote the set of probability distributions over the decisions $A(h_i^k)$ for player i at information set h_i^k. A behavior strategy for player i is an element $x_i \in \prod_{h_i^k \in \mathcal{H}_i} \Delta(A(h_i^k))$, where $x_i(h_i^k)$ denotes the probability distribution of the behavior strategy x_i over the available decisions at information set h_i^k.

Note that the space of admissible behavior strategies is much smaller than the space of admissible mixed strategies. To illustrate this, consider a player with K information sets and two possible decisions for each information set. The number of possible pure strategies would be 2^K, and thus the space of mixed strategies would be a probability simplex of dimension $2^K - 1$. In contrast, behavior strategies would require specifying probability distributions over two choices for each of K information sets, so the space of behavior strategies would be a product space of K probability simplices of dimension 1, totaling dimension K. One way of understanding this difference is that mixed strategies introduce correlated randomization across choices at different information sets, whereas behavior strategies introduce independent randomization across such choices.

For every behavior strategy, one can find an equivalent mixed strategy by computing the probabilities of every set of actions that result from the behavior strategy. The converse is not true for general games in extensive form. However, there is a special class of games for which the converse is true. In this class of games, players recall what actions they have selected previously and what information they knew previously. A formal definition of perfect recall is beyond the scope of this exposition but can be found in Hart (1991) and Kuhn (1953). The implication of perfect recall is summarized below:

Theorem 2 (Kuhn 1953) *Given a finite n-person game in which player i has perfect recall, for each mixed strategy μ_i for player i, there exists a corresponding behavior strategy x_i that*

is equivalent, where every player receives the same payoffs under both strategies μ_i and x_i.

This equivalence was extended by Aumann to infinite games (Aumann 1964). In dynamic games, it is common to assume that each player has perfect recall and thus solutions can be found in the smaller space of behavior strategies.

Computation of Equilibria

Algorithms for the computation of mixed-strategy Nash equilibria of static games can be extended to compute mixed-strategy Nash equilibria for games in extensive form when the pure strategies are enumerated as above. However, the number of pure strategies grows exponentially with the size of the extensive form tree, making these methods hard to apply. For two-person games in extensive form with perfect recall by both players, one can search for Nash equilibria in the much smaller space of behavior strategies. This was exploited in Koller et al. (1996) to obtain efficient linear complementarity problems for nonzero-sum games and linear programs for zero-sum games where the number of variables involved is linear in the number of internal decision nodes of the extensive form of the game. A more detailed overview of computation algorithms for Nash equilibria can be found in McKelvey and McLennan (1996). The Gambit Web site (McKelvey et al. 2010) provides software implementations of several techniques for computation of Nash equilibria in two- and n-person games.

An alternative approach to computing Nash equilibria for games in extensive forms is based on subgame decomposition, discussed next.

Subgames

Consider a game G in extensive form. A node c is a successor of a node n if there is a path in the game tree from n to c. Let h be a node in G that is not terminal and is the only node in its information set. Assume that if a node c is a successor of h, then every node in the information set containing c is also a successor of h. In this situation, one can define a subgame H of G with root node h, which consists of node h and
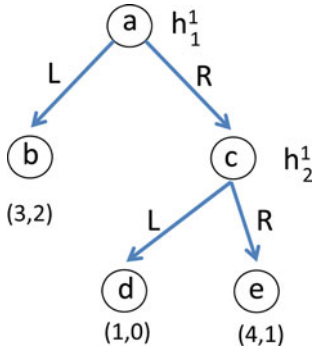
Dynamic Noncooperative Games, Fig. 2 Simple game with multiple Nash equilibria

its successors, connected by the same actions as in the original game G, with the payoffs and terminal vertices equal to those in G. This is the subgame that would be encountered by the players had the previous play of the game reached the state at node h. Since the information set containing node h contains no other node and all the information sets in the subgame contain only nodes in the subgame, then every player in the subgame knows that they are playing only in the subgame once h has been reached.

Figure 2 illustrates a game where every non-terminal node is contained in its own information set. This game contains a subgame rooted at node c. Note that the full game has two Nash equilibria in pure strategies: strategies (L, L) and (R, R). However, strategy (L, L) is inconsistent with how player 2 would choose its decision if it were to find itself at node c. This inconsistency arises because the Nash equilibria are defined in terms of strategies announced before the game is played and may not be a reasonable way to select decisions if unanticipated plays occur. When player 1 chooses L, node c should not be reached in the play of the game, and thus player 2 can choose L because it does not affect the expected payoff. One can define the concept of Nash equilibria that are sequentially consistent as follows.

Let x_i be behavior strategies for player i in game G. Denote the restriction of these strategies to the subgame H as x_i^H. This restriction describes the probabilities for choices of actions

for the information sets in H. Suppose the game G has a Nash equilibrium achieved by strategies $(x_1, \ldots, x_n)$. This Nash equilibrium is called *subgame perfect* (Selten 1975) if, for every subgame H of G, the strategies $(x_1^H, \ldots, x_n^H)$ are a Nash equilibrium for the subgame H. In the game in Fig. 2, there is only one subgame perfect equilibrium, which is (R, R). There are several other refinements of the Nash equilibrium concept to enforce sequential consistency, such as sequential equilibria (Kreps and Wilson 1982).

An important application of subgames is to compute subgame perfect equilibria by backward induction. The idea is to start with a subgame root node as close as possible to an outcome node (e.g., node c in Fig. 2). This small subgame can be solved for its Nash equilibria, to compute the equilibrium payoffs for each player. Then, in the original game G, the root node of the subgame can be replaced by an outcome node, with payoffs equal to the equilibrium payoffs in the subgame. This results in a smaller game, and the process can be applied inductively until the full game is solved. The subgame perfect equilibrium strategies can then be computed as the solution of the different subgames solved in this backward induction process. For Fig. 2, the subgame at c is solved by player 2 selecting R, with payoffs $(4, 1)$. The reduced new game has two choices for player 1, with best decision R. This leads to the overall subgame perfect equilibrium (R, R).

This backward induction procedure is similar to the dynamic programming approach to solving control problems. Backward induction was first used by Zermelo (1912) to analyze zero-sum games of perfect information such as chess. An extension of Zermelo's work by Kuhn (1953) establishes the following result:

Theorem 3 *Every finite game of perfect information has a subgame perfect Nash equilibrium in pure strategies.*

This result follows because, at each step in the backward induction process, the resulting subgame consists of a choice among finite actions for a single player, and thus a pure strategy achieves the maximal payoff possible.

Infinite Noncooperative Games

When the number of options available to players is infinite, game trees are no longer appropriate representations for the evolution of the game. Instead, one uses state space models with explicit models of actions and observations. A typical multistage model for the dynamics of such games is

$$x(t + 1) = f(x(t), u_1(t), \ldots, u_n(t), w(t), t),$$
$$t = 0, \ldots, T - 1 \qquad (2)$$

with initial condition $\underline{x}(0) = w(0)$, where $x(t)$ is the state of the game at stage t, and $u_1(t), \ldots, u_n(t)$ are actions selected by players $1, \ldots, n$ at stage t, and $w(t)$ is an action selected by Nature at stage t. The space of actions for each player are restricted at each time to infinite sets $A_i(t)$ with an appropriate topological structure, and the admissible state $x(t)$ at each time belongs to an infinite set X with a topological structure. In terms of Nature's actions, for each stage t, there is a probability distribution that specifies the choice of Nature's action $w(t)$, selected independently of other actions.

Equation (2) describes a play of the game, in terms of how different actions by players at the various stages evolve the state of the game. A play of the game is thus a history $\underline{h} = (x(0), u_1(0), \ldots, u_n(0), x(1), u_1(1), \ldots, u_n(1), \ldots, x(T))$. Associated with each play of the game is a set of real-valued functions $J_i(\underline{h})$ that indicates the payoff to player i in this play. This function is often assumed to be separable across the variables in each stage.

To complete the description of the extensive form, one must now introduce the available information to each player at each stage. Define observation functions

$$y_i(t) = g_i(x(t), v(t), t), \quad i = 1, \ldots, n$$

where $y_i(t)$ takes values in observations spaces which may be finite or infinite, and $v(t)$ are selected by Nature given their probability distributions, independent of other

selections. Define the information available for player i at stage t to be $I_i(t)$, a subset of $\{y_1(0), \ldots, y_n(0), \ldots, y_1(t), \ldots, y_n(t); u_1(0), \ldots, u_n(0), \ldots, u_1(t-1), \ldots, u_n(t-1)\}$. With this notation, strategies $\gamma_i(t)$ for player i are functions that map, at each stage, the available information $I_i(t)$ into admissible decisions $u_i(t) \in A_i(t)$. With appropriate measurability conditions, specifying a full set of strategies $\underline{\gamma} = (\gamma_1, \ldots \gamma_n)$ for each of the players induces a probability distribution on the plays of the game, which leads to the expected payoff $\mathcal{J}_i(\underline{\gamma})$. Nash equilibria are defined in identical fashion to (1).

Obtaining solutions of multistage games is a difficult task that depends on the ability to use the subgame decomposition techniques discussed previously. Such subgame decompositions are possible when games do not include actions by Nature and the payoff functions have a stagewise additive property. Under such cases, backward induction allows the construction of Nash equilibrium strategies through the recursive solutions of static infinite games, such as those discussed in the encyclopedia entry on static games.

Generalizations of multistage games to continuous time result in differential games, covered in two articles in the encyclopedia, but for the zero-sum case. Additional details on infinite dynamic noncooperative games, exploiting different models and information structures and studying the existence, uniqueness, or nonuniqueness of equilibria, can be found in Basar and Olsder (1982).

Conclusions

In this entry, we reviewed models for dynamic noncooperative games that incorporate temporal order of play and uncertainty as to what individual players know when they have to make decisions. Using these models, we defined solution concepts for the games and discussed algorithms for determining solution strategies for the players. Active directions of research include development of new solution concepts for dynamic games, new approaches to computation of game solutions, the study of games with a large

number of players, evolutionary games where players' greedy behavior evolves toward equilibrium strategies, and special classes of dynamic games such as Markov games and differential games. Several of these topics are discussed in other entries in the encyclopedia.

Cross-References

- ▶ Stochastic Dynamic Programming
- ▶ Stochastic Games and Learning
- ▶ Strategic Form Games and Nash Equilibrium

Bibliography

Aumann RJ (1964) Mixed and behavior strategies in infinite extensive games. In: Dresher M, Shapley LS, Tucker AW (eds) Advances in game theory. Princeton University Press, Princeton

Basar T, Olsder GJ (1982) Dynamic noncooperative game theory. Academic press, London

Hart S (1991) Games in extensive and strategic forms. In: Aumann R, Hart S (eds) Handbook of game theory with economic applications, vol 1. Elsevier, Amsterdam

Koller D, Megiddo N, von Stengel B (1996) Efficient computation of equilibria for extensive two-person games. Games Econ Behav 14:247–259

Kreps DM, Wilson RB (1982) Sequential equilibria. Econometrica 50:863–894

Kuhn HW (1953) Extensive games and the problem of information. In: Kuhn HW, Tucker AW (eds) Contributions to the theory of games, vol 2. Princeton University Press, Princeton

McKelvey RD, McLennan AM (1996) Computation of equilibria in finite games. In: Amman HM, Kendrick DA, Rust J (eds) Handbook of computational economics, vol 1. Elsevier, Amsterdam

McKelvey RD, McLennan AM, Turocy TL (2010) Gambit: software tools for game theory, version 0.2010.09.01. http://www.gambit-project.org

Nash J (1950) Equilibrium points in n-person games. Proc Natl Acad Sci 36(1):48–49

Nash J (1951) Non-cooperative games. Ann Math 54(2):286–295

Selten R (1975) Reexamination of the perfectness concept for equilibrium points in extensive games. Int J Game Theory 4(1):25–55

von Neumann J (1928) Zur theorie der gesellschaftsspiele. Mathematicsche Annale 100:295–320

Zermelo E (1912) Uber eine anwendung der mengenlehre auf die theoreie des schachspiels. In: Proceedings of the fifth international congress of mathematicians, Cambridge University Press, Cambridge, vol 2

Dynamic Positioning Control Systems for Ships and Underwater Vehicles

Asgeir J. Sørensen
Department of Marine Technology, Centre for Autonomous Marine Operations and Systems (AMOS), Norwegian University of Science and Technology, NTNU, Trondheim, Norway

Abstract

In 2012 the fleet of dynamically positioned (DP) ships and rigs was probably larger than 3,000 units, predominately operating in the offshore oil and gas industry. The complexity and functionality vary subject to the targeted marine operation, vessel concept, and risk level. DP systems with advanced control functions and redundant sensor, power, and thruster/propulsion configurations are designed in order to provide high-precision fault-tolerant control in safety-critical marine operations. The DP system is customized for the particular application with integration to other control systems, e.g., power management, propulsion, drilling, oil and gas production, offloading, crane operation, and pipe and cable laying. For underwater vehicles such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), DP functionality also denoted as *hovering* is implemented on several vehicles.

Keywords

Autonomous underwater vehicles (AUVs); Fault-tolerant control; Remotely operated vehicles (ROVs)

Introduction

The offshore oil and gas industry is the dominating market for DP vessels. The various offshore applications include offshore service

vessels, drilling rigs (semisubmersibles) and ships, shuttle tankers, cable and pipe layers, floating production, storage, and off-loading units (FPSOs), crane and heavy lift vessels, geological survey vessels, rescue vessels, and multipurpose construction vessels. DP systems are also installed on cruise ships, yachts, fishing boats, navy ships, tankers, and others.

A DP vessel is by the International Maritime Organization (IMO) and the maritime class societies (DNV GL, ABS, LR, etc.) defined as *a vessel that maintains its position and heading (fixed location denoted as stationkeeping or predetermined track) exclusively by means of active thrusters*. The DP system as defined by class societies is not only limited to the DP control system including computers and cabling. Position reference systems of various types measuring North-East coordinates (satellites, hydroacoustic, optics, taut wire, etc.), sensors (heading, roll, pitch, wind speed and direction, etc.), the power system, thruster and propulsion system, and independent joystick system are also essential parts of the DP system. In addition, the DP operator is an important element securing safe and efficient DP operations. Further development of human-machine interfaces, alarm systems, and operator decision support systems is regarded as top priority bridging advanced and complex technology to safe and efficient marine operations. Sufficient DP operator training is a part of this.

The thruster and propulsion system controlled by the DP control system is regarded as one of the main power consumers on the DP vessel. An important control system for successful integration with the power plant and the other power consumers such as drilling system, process system, heating, and ventilation system is the power and energy management system (PMS/EMS) balancing safety requirements and energy efficiency. The PMS/EMS controls the power generation and distribution and the load control of heavy power consumers. In this context both transient and steady-state behaviors are of relevance. A thorough understanding of the hydrodynamics, dynamics between coupled systems, load characteristics of the various power consumers, control system architecture, control layers, power

system, propulsion system, and sensors is important for successful design and operation of DP systems and DP vessels.

Thruster-assisted position mooring is another important stationkeeping application often used for FPSOs, drilling rigs, and shuttle tanker operations where the DP system has to be redesigned accounting for the effect of the mooring system dynamics. In thruster-assisted position mooring, the DP system is renamed to position mooring (PM) system. PM systems have been commercially available since the 1980s. While for DP-operated ships the thrusters are the sole source of the stationkeeping, the assistance of thrusters is only complementary to the mooring system. Here, most of the stationkeeping is provided by a deployed anchor system. In severe environmental conditions, the thrust assistance is used to minimize the vessel excursions and line tension by mainly increasing the damping in terms of velocity feedback control and adding a bias force minimizing the mean tensions of the most loaded mooring lines. Modeling and control of turret-anchored ships are treated in Strand et al. (1998) and Nguyen and Sørensen (2009).

Overview of DP systems including references can be found in Fay (1989), Fossen (2011), and Sørensen (2011). The scientific and industrial contributions since the 1960s are vast, and many research groups worldwide have provided important results. In Sørensen et al. (2012), the development of DP system for ROVs is presented.

Mathematical Modeling of DP Vessels

Depending on the operational conditions, the vessel models may briefly be classified into stationkeeping, low-velocity, and high-velocity models. As shown in Sørensen (2011) and Fossen (2011) and the references therein, different model reduction techniques are used for the various speed regimes. Vessel motions in waves are defined as seakeeping and will here apply both for stationkeeping (zero speed) and forward speed. DP vessels or PM vessels can in general be regarded as stationkeeping and low-velocity or low Froude

number applications. This assumption will particularly be used in the formulation of mathematical models used in conjunction with the controller design. It is common to use a two-time scale formulation by separating the total model into a low-frequency (LF) model and a wave-frequency (WF) model (seakeeping) by superposition. Hence, the total motion is a sum of the corresponding LF and the WF components. The WF motions are assumed to be caused by first-order wave loads. Assuming small amplitudes, these motions will be well represented by a linear model. The LF motions are assumed to be caused by second-order mean and slowly varying wave loads, current loads, wind loads, mooring (if any), and thrust and rudder forces and moments.

For underwater vehicles operating below the wave zone, estimated to be deeper than half the wavelength, the wave loads can be disregarded and of course the effect of the wind loads as well.

Modeling Issues

The mathematical models may be formulated in two complexity levels:

- *Control plant model* is a simplified mathematical description containing only the main physical properties of the process or plant. This model may constitute a part of the controller. The control plant model is also used in analytical stability analysis based on, e.g., Lyapunov stability.
- *Process plant model* or simulation model is a comprehensive description of the actual process and should be as detailed as needed. The main purpose of this model is to simulate the real plant dynamics. The process plant model is used in numerical performance and robustness analysis and testing of the control systems. As shown above, the process plant models may be implemented for off-line or real-time simulation (e.g., HIL testing; see Johansen et al. 2007) purposes defining different requirements for model fidelity.

Kinematics

The relationship between the Earth-fixed position and orientation of a floating structure and its body-fixed velocities is

$$
\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1(\boldsymbol{\eta}_2) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J}_2(\boldsymbol{\eta}_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} \quad (1)
$$

The vectors defining the Earth-fixed vessel position ($\boldsymbol{\eta}_1$) and orientation ($\boldsymbol{\eta}_2$) using Euler angles and the body-fixed translation ($\boldsymbol{v}_1$) and rotation ($\boldsymbol{v}_2$) velocities are given by

$$
\boldsymbol{\eta}_1 = \begin{bmatrix} x & y & z \end{bmatrix}^T, \boldsymbol{\eta}_2 = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T,
$$
$$
\boldsymbol{v}_1 = \begin{bmatrix} u & v & w \end{bmatrix}^T, \boldsymbol{v}_2 = \begin{bmatrix} p & q & r \end{bmatrix}^T. \quad (2)
$$

The rotation matrix $\mathbf{J}_1(\boldsymbol{\eta}_2) \in \mathbf{SO}(3)$ and the velocity transformation matrix $\mathbf{J}_2(\boldsymbol{\eta}_2) \in \mathbb{R}^{3\times3}$ are defined in Fossen (2011). For ships, if only surge, sway and yaw (3DOF) are considered, the kinematics and the state vectors are reduced to

$$
\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{v}, \text{ or } \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix}
$$
$$
= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}. \quad (3)
$$

Process Plant Model: Low-Frequency Motion

The 6-DOF LF model formulation is based on Fossen (2011) and Sørensen (2011). The equations of motion for the nonlinear LF model of a floating vessel are given by

$$
\mathbf{M}\dot{\boldsymbol{v}} + \mathbf{C}_{RB}(\boldsymbol{v})\boldsymbol{v} + \mathbf{C}_A(\boldsymbol{v}_r)\boldsymbol{v}_r + \mathbf{D}(\boldsymbol{v}_r) + \mathbf{G}(\boldsymbol{\eta})
$$
$$
= \boldsymbol{\tau}_{\text{wave2}} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{thr}} + \boldsymbol{\tau}_{\text{moor}}, \quad (4)
$$

where $\mathbf{M} \in \mathbb{R}^{6\times6}$ is the system inertia matrix including added mass; $\mathbf{C}_{RB}(\boldsymbol{v}) \in \mathbb{R}^{6\times6}$ and $\mathbf{C}_A(\boldsymbol{v}_r) \in \mathbb{R}^{6\times6}$ are the skew-symmetric Coriolis and centripetal matrices of the rigid body and the added mass; $\mathbf{G}(\boldsymbol{\eta}) \in \mathbb{R}^6$ is the generalized restoring vector caused by the mooring lines (if any), buoyancy, and gravitation; $\boldsymbol{\tau}_{\text{thr}} \in \mathbb{R}^6$ is the control vector consisting of forces and moments produced by the thruster system; $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave2}} \in \mathbb{R}^6$ are the wind and second-order wave load vectors, respectively.

The damping vector may be divided into linear and nonlinear terms according to

$$\mathbf{D}(\boldsymbol{v}_r) = \mathbf{d}_L(\boldsymbol{v}_r, \kappa)\boldsymbol{v}_r + \mathbf{d}_{NL}(\boldsymbol{v}_r, \gamma_r)\boldsymbol{v}_r, \quad (5)$$

where $\boldsymbol{v}_r \in \mathbb{R}^6$ is the relative velocity vector between the current and the vessel. The nonlinear damping, $\mathbf{d}_{NL}$, is assumed to be caused by turbulent skin friction and viscous eddy-making, also denoted as vortex shedding (Faltinsen 1990). The strictly positive linear damping matrix $\mathbf{d}_L \in \mathbb{R}^{6 \times 6}$ is caused by linear laminar skin friction and is assumed to vanish for increasing speed according to

$$\mathbf{d}_L(\boldsymbol{v}_r, \kappa) = \begin{bmatrix} X_{u_r} e^{-\kappa|u_r|} & .. & X_r e^{-\kappa|r|} \\ .. & .. & .. \\ N_{u_r} e^{-\kappa|u_r|} & .. & N_r e^{-\kappa|r|} \end{bmatrix}, \quad (6)$$

where κ is a positive scaling constant such that $\kappa \in \mathrm{R}^+$.

Process Plant Model: Wave-Frequency Motion

The coupled equations of the WF motions in surge, sway, heave, roll, pitch, and yaw are assumed to be linear and can be formulated as

$$\begin{aligned} \mathbf{M}(\omega)\ddot{\boldsymbol{\eta}}_{Rw} + \mathbf{D}_p(\omega)\dot{\boldsymbol{\eta}}_{Rw} + \mathbf{G}\boldsymbol{\eta}_{Rw} &= \boldsymbol{\tau}_{\text{wave1}}, \\ \dot{\boldsymbol{\eta}}_w &= \mathbf{J}(\bar{\boldsymbol{\eta}}_2)\dot{\boldsymbol{\eta}}_{Rw}, \end{aligned}$$
$$(7)$$

where $\boldsymbol{\eta}_{Rw} \in \mathbb{R}^6$ is the WF motion vector in the hydrodynamics frame. $\boldsymbol{\eta}_w \in \mathbb{R}^6$ is the WF motion vector in the Earth-fixed frame. $\boldsymbol{\tau}_{\text{wave1}} \in \mathbb{R}^6$ is the first-order wave excitation vector, which will be modified for varying vessel headings relative to the incident wave direction. $\mathbf{M}(\omega) \in \mathbb{R}^{6 \times 6}$ is the system inertia matrix containing frequency dependent added mass coefficients in addition to the vessel's mass and moment of inertia. $\mathbf{D}_p(\omega) \in \mathbb{R}^{6 \times 6}$ is the wave radiation (potential) damping matrix. The linearized restoring coefficient matrix $\mathbf{G} \in \mathbb{R}^{6 \times 6}$ is due to gravity and buoyancy affecting heave, roll, and pitch only. For anchored vessels, it is assumed that the mooring system will not influence the WF motions.

Remark 1 Generally, a time domain equation cannot be expressed with frequency domain coefficient – ω. However, this is a common used formulation denoted as a pseudo-differential equation. An important feature of the added mass terms and the wave radiation damping terms is the memory effects, which in particular are important to consider for nonstationary cases, e.g., rapid changes of heading angle. Memory effects can be taken into account by introducing a convolution integral or a so-called retardation function (Newman 1977) or state space models as suggested by Fossen (2011).

Control Plant Model

For the purpose of controller design and analysis, it is convenient to apply model reduction and derive a LF and WF control plant model in surge, sway, and yaw about zero vessel velocity according to

$$\dot{\mathbf{p}}_w = \mathbf{A}_{pw}\mathbf{p}_w + \mathbf{E}_{pw}\mathbf{w}_{pw}, \quad (8)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{v}, \quad (9)$$

$$\dot{\mathbf{b}} = -\mathbf{T}_b\mathbf{b} + \mathbf{E}_b\mathbf{w}_b, \quad (10)$$

$$\mathbf{M}\dot{\boldsymbol{v}} = -\mathbf{D}_L\boldsymbol{v} + \mathbf{R}^T(\psi)\mathbf{b} + \boldsymbol{\tau}, \quad (11)$$

$$\dot{\omega}_p = 0, \quad (12)$$

$$\mathbf{y} = \left[\left(\boldsymbol{\eta} + \mathbf{C}_{pw}\mathbf{p}_w \right)^T \quad \omega_p \right]^T, \quad (13)$$

where $\omega_p \in \mathbb{R}$ is the peak frequency of the waves (PFW). The estimated PFW can be calculated by spectral analysis of the pitch and roll measurements assumed to dominantly oscillate at the peak wave frequency. In the spectral analysis, the discrete Fourier transforms of the measured roll and pitch, which are collected through a period of time, are done by taking the n-point fast Fourier transform (FFT). The PFW may be found to be the frequency at which the power spectrum is maximal. The assumption $\dot{\omega}_p = 0$ is valid for slowly varying sea state. You can also find the wave frequency using nonlinear observers/EKF and signal processing techniques. It is assumed that the second-order linear model is sufficient to describe the first-order wave-induced motions, and then $\mathbf{p}_w \in \mathbb{R}^6$ is the state of the WF model. $\mathbf{A}_{pw} \in \mathbb{R}^{6 \times 6}$ is assumed Hurwitz and describes the first-order wave-induced motion as a

mass-damper-spring system. $\mathbf{w}_{pw} \in \mathbb{R}^3$ is a zero-mean Gaussian white noise vector. $\mathbf{y}$ is the measurement vector. The WF measurement matrix $\mathbf{C}_{pw} \in \mathbb{R}^{3\times6}$ and the disturbance matrix $\mathbf{E}_{pw} \in \mathbb{R}^{6\times3}$ are formulated as

$$\mathbf{C}_{pw} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \mathbf{E}_{pw}^T = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{K}_w^T \end{bmatrix}^T. \tag{14}$$

Here, a 3-DOF model is assumed adopting the notation in (3) such that $\boldsymbol{\eta} \in \mathbb{R}^3$ and $\boldsymbol{\upsilon} \in \mathbb{R}^3$ are the LF position vector in the Earth-fixed frame and the LF velocity vector in the body-fixed frame, respectively. $\mathbf{M} \in \mathbb{R}^{3\times3}$ and $\mathbf{D}_L \in \mathbb{R}^{3\times3}$ are the mass matrix including hydrodynamic added mass and linear damping matrix, respectively. The bias term accounting for unmodeled affects and slowly varying disturbances $\mathbf{b} \in \mathbb{R}^3$ is modeled as Markov processes with positive definite diagonal matrix $\mathbf{T}_b \in \mathbb{R}^{3\times3}$ of time constants. If $\mathbf{T}_b$ is removed, a Wiener process is used. $\mathbf{w}_b \in \mathbb{R}^3$ is a bounded disturbance vector, and $\mathbf{E}_b \in \mathbb{R}^{3\times3}$ is a disturbance scaling matrix. $\boldsymbol{\tau} \in \mathbb{R}^3$ is the control force. As mentioned later in the paper, the proposed model reduction considering only horizontal motions may create problems conducting DP operations of structures with low waterplane area such as semisubmersibles. More details can be found in Sørensen (2011) and Fossen (2011) and the references therein.

For underwater vehicles, 6-DOF model should be used. For underwater vehicles with self-stabilizing roll and pitch, a 4-DOF model with surge, sway, yaw, and heave may be used; see Sørensen et al. (2012).

Control Levels and Integration Aspects

The real-time control hierarchy of a marine control system (Sørensen 2005) may be divided into three levels: *the guidance system and local optimization*, *the high-level plant control* (e.g., DP controller including thrust allocation), and *the low-level thruster control*. The DP control system consists of several modules as indicated in Fig. 1:

- *Signal processing* for analysis and testing of the individual signals including voting and weighting when redundant measurements are available. Ensuring robust and fault-tolerant control proper diagnostics and change detection algorithms is regarded as maybe one of the most important research areas. For an overview of the field, see Basseville and Nikiforov (1993) and Blanke et al. (2003).

- *Vessel observer* for state estimation and wave filtering. In case of lost sensor signals, the predictor is used to provide dead reckoning, which is required by class societies. Prediction error which is the deviation between the measurements and the estimated measurements is also one important barrier in the failure detection.

- *Feedback control law* is often of multivariable PID type, where feedback is produced from the estimated low-frequency (LF) position and heading deviations and estimated LF velocities.

- *Feedforward control law* is normally the wind force and moment. For the different applications (pipe laying, ice operations, position mooring), tailor-made feedforward control functions are also used.

- *Guidance system with reference models* is needed in achieving a smooth transition between setpoints. In the most basic case, the operator specifies a new desired position and heading, and a reference model generates smooth reference trajectories/paths for the vessel to follow. A more advanced guidance system involves way-point tracking functionality with optimal path planning.

- *Thrust allocation* computes the force and direction commands to each thruster device based on input from the resulting feedback and feedforward controllers. The low-level thruster controllers will then control the propeller pitch, speed, torque, and power.

- *Model adaptation* provides the necessary corrections of the vessel model and the controller settings subject to changes in the vessel draft, wind area, and variations in the sea state.

- *Power management system* performs diesel engine control, power generation management with frequency and voltage monitoring, active and passive load sharing, and load dependent start and stop of generator sets.
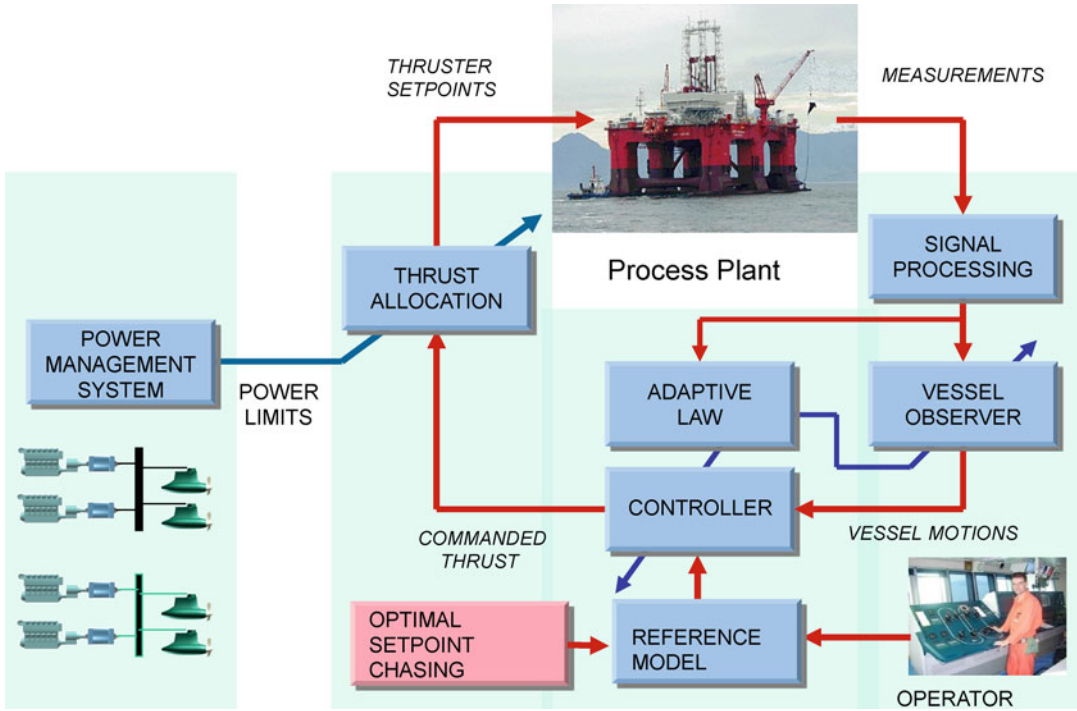
Dynamic Positioning Control Systems for Ships and Underwater Vehicles, Fig. 1   Controller structure

DP Controller

In the 1960s the first DP system was introduced for horizontal modes of motion (surge, sway, and yaw) using single-input single-output PID control algorithms in combination with low-pass and/or notch filters. In the 1970s more advanced output control methods based on multivariable optimal control and Kalman filter theory were proposed by Balchen et al. (1976) and later refined in Sælid et al. (1983); Grimble and Johnson (1988); and others as referred to in Sørensen (2011). In the 1990s nonlinear DP controller designs were proposed by several research groups; for an overview see Strand et al. (1998), Fossen and Strand (1999), Pettersen and Fossen (2000), and Sørensen (2011). Nguyen et al. (2007) proposed the design of hybrid controller for DP from calm to extreme sea conditions.

Plant Control

By copying the control plant model (8)–(13) and adding an injection term, a passive observer may

be designed. A nonlinear output horizontal-plane positioning feedback controller of PID type may be formulated as

$$\boldsymbol{\tau}_{PID} = -\mathbf{R}_e^{\mathbf{T}}\mathbf{K_p}\mathbf{e} - \mathbf{R}_e^{\mathbf{T}}\mathbf{K_{p3}}\mathbf{f(e)} - \mathbf{K_d}\tilde{\boldsymbol{v}} - \mathbf{R}^{\mathbf{T}}\mathbf{K_i}\mathbf{z}, \tag{15}$$

where $\mathbf{e} \in \mathbb{R}^3$ is the position and heading deviation vector, $\tilde{\boldsymbol{v}} \in \mathbb{R}^3$ is the velocity deviation vector, $\mathbf{z} \in \mathbb{R}^3$ is the integrator states, and $\mathbf{f(e)}$ is a third-order stiffness term defined as

$$\begin{aligned}
\mathbf{e} &= [e_1, e_2, e_3]^T = \mathbf{R}^T(\phi_d)(\widehat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d), \\
\tilde{\boldsymbol{v}} &= \widehat{\boldsymbol{v}} - \mathbf{R}^T(\phi_d)\boldsymbol{\eta}_d, \\
\dot{\mathbf{z}} &= \widehat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d, \\
\mathbf{R}_e &= \mathbf{R}(\phi - \phi_d) = \mathbf{R}^T(\phi_d)\mathbf{R}(\phi), \\
\mathbf{f(e)} &= [e_1^3, e_2^3, e_3^3]^T.
\end{aligned}$$

Experience from implementation and operations of real DP control systems has shown that ϕ_d in the calculation of the error vector $\mathbf{e}$ generally gives a better performance with less noisy signals than using ϕ. However, this is only valid
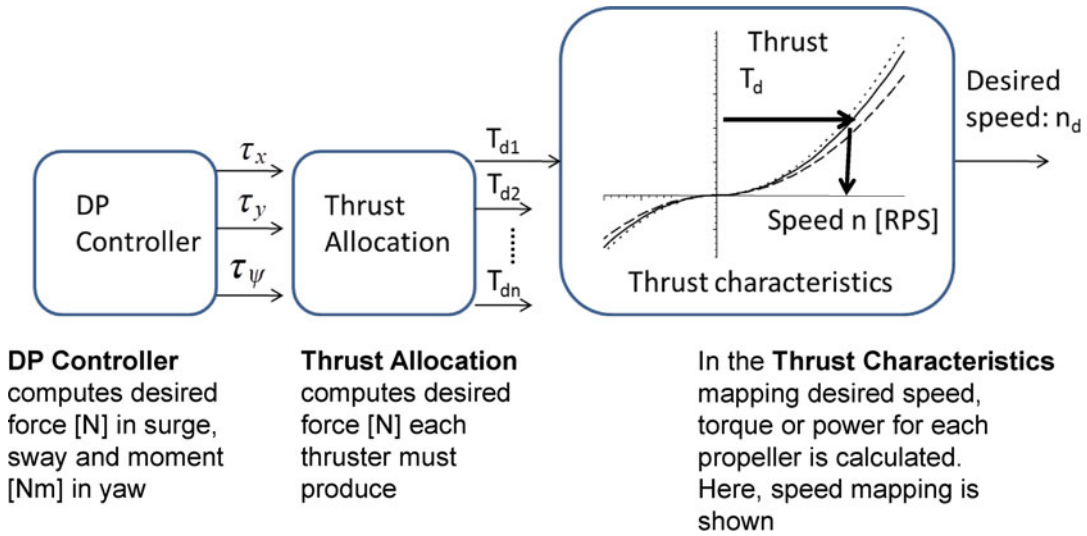
Dynamic Positioning Control Systems for Ships and Underwater Vehicles, Fig. 2 Thrust allocation

under the assumption that the vessel maintains its desired heading with small deviations. As the DP capability for ships is sensitive to the heading angle, i.e., minimizing the environmental loads, heading control is prioritized in case of limitations in the thrust capacity. This feature is handled in the thrust allocation.

An advantage of this is the possibility to reduce the first-order proportional gain matrix, resulting in reduced dynamic thruster action for smaller position and heading deviations. Moreover, the third-order restoring term will make the thrusters to work more aggressive for larger deviations. $\mathbf{K}_p$, $\mathbf{K}_{p3}$, $\mathbf{K}_d$, and $\mathbf{K}_i \in \mathbb{R}^{3 \times 3}$ are the nonnegative controller gain matrices for proportional, third-order restoring, derivative, and integrator controller terms, respectively, found by appropriate controller synthesis methods.

For small-waterplane-area marine vessels such as semisubmersibles, often used as drilling rigs, Sørensen and Strand (2000) proposed a DP control law with the inclusion of roll and pitch damping according to

$$\tau_{rpd} = - \begin{bmatrix} 0 & g_{xq} \\ g_{yp} & 0 \\ g_{\phi p} & 0 \end{bmatrix} \begin{bmatrix} \widehat{p} \\ \widehat{q} \end{bmatrix}, \qquad (16)$$

where $\widehat{p}$ and $\hat{q}$ are the estimated pitch and roll angular velocities. The resulting positioning control law is written as

$$\tau = \tau_{wFF} + \tau_{PID} + \tau_{rpd}, \qquad (17)$$

where $\tau_{wFF} \in \mathbb{R}^3$ is the wind feedforward control law.

Thrust allocation or control allocation (Fig. 2) is the mapping between plant and actuator control. It is assumed to be a part of the plant control. The DP controller calculated the desired force in surge and sway and moment in yaw. Dependent on the particular thrust configuration with installed and enabled propellers, tunnel thrusters, azimuthing thrusters, and rudders, the allocation is a nontrivial optimization problem calculating the desired thrust and direction for each enabled thruster subject to various constraints such as thruster ratings, forbidden sectors, and thrust efficiency. References on thrust allocation are found in Johansen and Fossen (2013).

In Sørensen and Smogeli (2009), torque and power control of electrically driven marine propellers are shown. Ruth et al. (2009) proposed anti-spin thrust allocation, and Smogeli et al.

(2008) and Smogeli and Sørensen (2009) presented the concept of anti-spin thruster control.

Cross-References

▶ Control of Ship Roll Motion
▶ Fault-Tolerant Control
▶ Mathematical Models of Ships and Underwater Vehicles
▶ Motion Planning for Marine Control Systems
▶ Underactuated Marine Control Systems

Bibliography

Balchen JG, Jenssen NA, Sælid S (1976) Dynamic positioning using Kalman filtering and optimal control theory. In: IFAC/IFIP symposium on automation in offshore oil field operation, Amsterdam, pp 183–186

Basseville M, Nikiforov IV (1993) Detection of abrupt changes: theory and application. Prentice-Hall, Englewood Cliffs. ISBN:0-13-126780-9

Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2003) Diagnostics and fault-tolerant control. Springer, Berlin

Faltinsen OM (1990) Sea loads on ships and offshore structures. Cambridge University Press, Cambridge

Fay H (1989) Dynamic positioning systems, principles, design and applications. Editions Technip, Paris. ISBN:2-7108-0580-4

Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

Fossen TI, Strand JP (1999) Passive nonlinear observer design for ships using Lyapunov methods: experimental results with a supply vessel. Automatica 35(1):3–16

Grimble MJ, Johnson MA (1988) Optimal control and stochastic estimation: theory and applications, vols 1 and 2. Wiley, Chichester

Johansen TA, Fossen TI (2013) Control allocation—a survey. Automatica 49(5):1087–1103

Johansen TA, Sørensen AJ, Nordahl OJ, Mo O, Fossen TI (2007) Experiences from hardware-in-the-loop (HIL) testing of dynamic positioning and power management systems. In: OSV Singapore, Singapore

Newman JN (1977) Marine hydrodynamics. MIT, Cambridge

Nguyen TD, Sørensen AJ (2009) Setpoint chasing for thruster-assisted position mooring. IEEE J Ocean Eng 34(4):548–558

Nguyen TD, Sørensen AJ, Quek ST (2007) Design of hybrid controller for dynamic positioning from calm to extreme sea conditions. Automatica 43(5):768–785

Pettersen KY, Fossen TI (2000) Underactuated dynamic positioning of a ship – experimental results. IEEE Trans Control Syst Technol 8(4):856–863

Ruth E, Smogeli ØN, Perez T, Sørensen AJ (2009) Anti-spin thrust allocation for marine vessels. IEEE Trans Control Syst Technol 17(6):1257–1269

Sælid S, Jenssen NA, Balchen JG (1983) Design and analysis of a dynamic positioning system based on Kalman filtering and optimal control. IEEE Trans Autom Control 28(3):331–339

Smogeli ØN, Sørensen AJ (2009) Antispin thruster control for ships. IEEE Trans Control Syst Technol 17(6):1362–1375

Smogeli ØN, Sørensen AJ, Minsaas KJ (2008) The concept of anti-spin thruster control. Control Eng Pract 16(4):465–481

Strand JP, Fossen TI (1999) Nonlinear passive observer for ships with adaptive wave filtering. In: Nijmeijer H, Fossen TI (eds) New directions in nonlinear observer design. Springer, London, pp 113–134

Strand JP, Sørensen AJ, Fossen TI (1998) Design of automatic thruster assisted position mooring systems for ships. Model Identif Control 19(2):61–75

Sørensen AJ (2005) Structural issues in the design and operation of marine control systems. Annu Rev Control 29(1):125–149

Sørensen AJ (2011) A survey of dynamic positioning control systems. Annu Rev Control 35:123–136.

Sørensen AJ, Smogeli ØN (2009) Torque and power control of electrically driven marine propellers. Control Eng Pract 17(9):1053–1064

Sørensen AJ, Strand JP (2000) Positioning of small-waterplane-area marine constructions with roll and pitch damping. Control Eng Pract 8(2):205–213

Sørensen AJ, Dukan F, Ludvigsen M, Fernandez DA, Candeloro M (2012) Development of dynamic positioning and tracking system for the ROV Minerva. In: Roberts G, Sutton B (eds) Further advances in unmanned marine vehicles. IET, London, pp 113–128. Chapter 6

E

Economic Model Predictive Control

David Angeli
Department of Electrical and Electronic
Engineering, Imperial College London,
London, UK
Dipartimento di Ingegneria dell'Informazione,
University of Florence, Italy

Abstract

Economic model predictive control (EMPC) is a variant of model predictive control aimed at maximization of system's profitability. It allows one to explicitly deal with hard and average constraints on system's input and output variables as well as with nonlinearity of dynamics. We provide basic definitions and concepts of the approach and highlight some promising research directions.

Keywords

Constrained systems; Dynamic programming; Profit maximization

Introduction

Most control tasks involve some kind of economic optimization. In classical linear quadratic (LQ) control, for example, this is cast as a trade-off between control effort and tracking performance. The designer is allowed to settle such a trade-off by suitably tuning weighting parameters of an otherwise automatic design procedure.

When the primary goal of a control system is profitability rather than tracking performance, a suboptimal approach has often been devised, namely, a hierarchical separation is enforced between the economic optimization layer and the dynamic real-time control layer.

In practice, while set points are computed by optimizing economic revenue among all equilibria fulfilling the prescribed constraints, the task of the real-time control layer is simply to drive (basically as fast as possible) the system's state to the desired set-point value.

Optimal control or LQ control may be used to achieve the latter task, possibly in conjunction with model predictive control (MPC), but the actual economics of the plant are normally neglected at this stage.

The main benefits of this approach are twofold:

1. Reduced computational complexity with respect to infinite-horizon dynamical programming
2. Stability robustness in the face of uncertainty, normally achieved by using some form of robust control in the real-time control layer

The hierarchical approach, however, is suboptimal in two respects:

1. First of all, given nonlinearity of the plant's dynamics and/or nonconvexity of the

functions characterizing the economic revenue, there is no reason why the most profitable regime should be an equilibrium.

2. Even when systems are most profitably operated at equilibrium, transient costs are totally disregarded by the hierarchical approach and this may be undesirable if the time constants of the plant are close enough to the time scales at which set point's variations occur.

Economic model predictive control seeks to remove these limitations by directly using the economic revenue in the stage cost and by the formulation of an associated dynamic optimization problem to be solved online in a receding horizon manner. It was originally developed by Rawlings and co-workers, in the context of linear control systems subject to convex constraints as an effective technique to deal with infeasible set points (Rawlings et al. 2008) (in contrast to the classical approach of redesigning a suitable quadratic cost that achieves its minimum at the closest feasible equilibrium). Preserving the original cost has the advantage of slowing down convergence to such an equilibrium when the transient evolution occurs in a region where the stage cost is better than at steady state. Stability and convergence issues are at first analyzed, thanks to convexity and for the special case of linear systems only. Subsequently Diehl introduced the notion of rotated cost (see Diehl et al. 2011) that allowed a Lyapunov interpretation of stability criteria and paved the way for the extension to general dissipative nonlinear systems (Angeli et al. 2012).

Economic MPC Formulation

In order to describe the most common versions of economic MPC, assume that a discrete-time finite-dimensional model of state evolution is available for the system to be controlled:

$$x^+ = f(x, u) \qquad (1)$$

where $x \in X \subset \mathbb{R}^n$ is the state variable, $u \in U \subset \mathbb{R}^m$ is the control input, and $f : X \times U \to X$ is a continuous map which computes the next state value, given the current one and the value of

the input. We also assume that $\mathbb{Z} \subset X \times U$ is a compact set which defines the (possibly coupled) state/input constraints that need to hold pointwise in time:

$$(x(t), u(t)) \in \mathbb{Z} \qquad \forall t \in \mathbb{N}. \qquad (2)$$

In order to introduce a measure of economic performance, to each feasible state/input pair $(x, u) \in \mathbb{Z}$, we associate the instantaneous net cost of operating the plant at that state when feeding the specified control input:

$$\ell(x, u) \; : \; \mathbb{Z} \to \mathbb{R}. \qquad (3)$$

The function ℓ (which we assume to be continuous) is normally referred to as stage cost and together with actuation and/or inflow costs should also take into account the profits associated to possible output/outflows of the system. Let (x^*, u^*) denote the best equilibrium/control input pair associated to (3) and (2), namely,

$$\begin{aligned} \ell(x^*, u^*) = \min_{x,u} \ell(x, u) \\ \text{subject to} \\ (x, u) \in \mathbb{Z} \\ x = f(x, u) \end{aligned} \qquad (4)$$

Notice that, unlike in tracking MPC, it is not assumed here that

$$\ell(x^*, u^*) \le \ell(x, u) \quad \forall(x, u) \in \mathbb{Z}. \qquad (5)$$

This is, technically speaking, the main point of departure between economic MPC and tracking MPC.

As there is no natural termination time to operation of a system, our goal would be to optimize the infinite-horizon cost functional:

$$\sum_{t \in \mathbb{N}} \ell(x(t), u(t)) \qquad (6)$$

possibly in an average sense (or by introducing some discounting factor to avoid infinite costs) and subject to the dynamic/operational constraints (1) and (2).

To make the problem computationally more tractable and yet retain some of the desirable economic benefits of dynamic programming, (6) is truncated to the following cost functional:

$$J(\mathbf{z}, \mathbf{v}) = \sum_{k=0}^{N-1} \ell(z(k), v(k)) + V_f(z(N)) \quad (7)$$

where $\mathbf{z} = [z(0), z(1), \ldots, z(N)] \in X^{N+1}, \mathbf{v} = [v(0), v(1), \ldots, v(N-1)] \in U^N$ and $V_f \colon X \to \mathbb{R}$ is a terminal weighting function whose properties will be specified later.

The virtual state/control pair $(\mathbf{z}^*, \mathbf{v}^*)$ at time t is the solution (which for the sake of simplicity we assume to be unique) of the following optimization problem:

$$\begin{aligned} V(x(t)) &= \min_{\mathbf{z}, \mathbf{v}} J(\mathbf{z}, \mathbf{v}) \\ &\text{subject to} \\ &z(k+1) = f(z(k), v(k)) \\ &(z(k), v(k)) \in \mathbb{Z} \\ &\text{for } k \in \{0, 1, \ldots, N-1\} \\ &z(0) = x(t), \ z(N) \in \mathbb{X}_f. \end{aligned} \quad (8)$$

Notice that $z(0)$ is initialized at the value of the current state $x(t)$. Thanks to this fact, $\mathbf{z}^*$ and $\mathbf{v}^*$ may be seen as functions of the current state $x(t)$. At the same time, $z(N)$ is constrained to belong to the compact set $\mathbb{X}_f \subset X$ whose properties will be detailed in the next paragraph.

As customary in model predictive control, a state-feedback law is defined by applying the first virtual control to the plant, that is, by letting $u(t) = v^*(0)$ and restating, at the subsequent time instant, the same optimization problem from initial state $x(t+1)$ which, in the case of exact match between plant and model, can be computed as $f(x(t), u(t))$.

In the next paragraph, we provide details on how to design the "terminal ingredients" (namely, V_f and $\mathbb{X}_f$) in order to endow the basic algorithm (8) with important features such as recursive feasibility and a certain degree of average performance and/or stability).

Hereby it is worth pointing out how, in the context of economic MPC, it makes sense to treat, together with pointwise-in-time constraints, asymptotic average constraints on specified input/output variables. In tracking applications, where the control algorithm guarantees asymptotic convergence of the state to a feasible set point, the average asymptotic value of all input/output variables necessarily matches that of the corresponding equilibrium/control input pair. In economic MPC, the asymptotic regime resulting in closed loop may, in general, fail to be an equilibrium; therefore, it might be of interest to impose average constraints on system's inflows and outflows which are more stringent than those indirectly implied by the fulfillment of (2). To this end, let the system's output be defined as

$$y(t) = h(x(t), u(t)) \quad (9)$$

with $h(x, u) : \mathbb{Z} \to \mathbb{R}^p$, a continuous map, and consider the convex compact set $\mathbb{Y}$. We may define the set of asymptotic averages of a bounded signal y as follows:

$$\mathrm{Av}[y] = \left\{ \eta \in \mathbb{R}^p : \exists \{t_n\}_{n=1}^{\infty} : t_n \to \infty \text{ as } n \to \infty \right.$$
$$\left. \text{and } \eta = \lim_{n \to \infty} \left(\sum_{k=0}^{t_n-1} y(k) \right) / t_n \right\}$$

Notice that for converging signals, or even for periodic ones, $\mathrm{Av}[y]$ always is a singleton but may fail to be such for certain oscillatory regimes. An asymptotic average constraint can be expressed as follows:

$$\mathrm{Av}[y] \subseteq \mathbb{Y} \quad (10)$$

where y is the output signal as defined in (9).

Basic Theory

The main theoretical results in support of the approach discussed in the previous paragraph are discussed below. Three fundamental aspects are treated:

- Recursive feasibility and constraint satisfaction
- Asymptotic performance
- Stability and convergence

Feasibility and Constraints

The departing point of most model predictive control techniques is to ensure recursive feasibility, namely, the fact that feasibility of the problem (8) at time 0 implies feasibility at all subsequent times, provided there is no mismatch between the true plant and its model (1). This is normally achieved by making use of a suitable notion of control invariant set which is used as a terminal constraint in (8). Economic model predictive control is not different in this respect, and either one of the following set of assumptions is sufficient to ensure recursive feasibility:

1. **Assumption 1: Terminal constraint**

$$\mathbb{X}_f = \{x^*\} \qquad V_f = 0$$

2. **Assumption 2: Terminal penalty function**

There exists a continuous map $\kappa : \mathbb{X}_f \to U$ such that

$$(x, \mathbb{K}(x)) \in \mathbb{Z} \qquad \forall x \in \mathbb{X}_f$$
$$f(x, \mathbb{K}(x)) \in \mathbb{X}_f \qquad \forall x \in \mathbb{X}_f$$

The following holds:

Theorem 1 *Let $x(0)$ be a feasible state for (8) and assume that either Assumption 1 or 2 hold. Then, the closed-loop trajectory $x(t)$ resulting from receding horizon implementation of the feedback $u(t) = v^*(0)$ is well defined for all $t \in \mathbb{N}$ (i.e., $x(t)$ is a feasible initial state of (8) for all $t \in \mathbb{N}$) and the resulting closed-loop variables $(x(t), u(t))$ fulfill the constraints in (2).*

The proof of this Theorem can be found in Angeli et al. (2012) and Amrit et al. (2011), for instance. When constraints on asymptotic averages are of interest, the optimization problem (8) can be augmented by the following constraints:

$$\sum_{k=0}^{N-1} h(z(k), v(k)) \in \mathbb{Y}_t \qquad (11)$$

provided $\mathbb{Y}_t$ is recursively defined as

$$\mathbb{Y}_{t+1} = \mathbb{Y}_t \oplus \mathbb{Y} \oplus \{-h(x(t), u(t))\} \qquad (12)$$

where $\oplus$ denotes Pontryagin's set sum. ($A \oplus B := \{c : \exists a \in A, \exists b \in B : c = a + b\}$) The sequence is initialized as $\mathbb{Y}_0 = N\mathbb{Y} \oplus \mathbb{Y}_{00}$ where $\mathbb{Y}_{00}$ is an arbitrary compact set in $\mathbb{R}^p$ containing 0 in its interior. The following result can be proved.

Theorem 2 *Consider the optimization problem (8) with additional constraints (11), and assume that $x(0)$ is a feasible initial state. Then, provided a terminal equality constraint is adopted, the closed-loop solution $x(t)$ is well defined and feasible for all $t \in \mathbb{N}$ and the resulting closed-loop variable $y(t) = h(x(t), u(t))$ fulfills the constraint (10).*

Extending average constraints to the case of economic MPC with terminal penalty function is possible but outside the scope of this brief tutorial. It is worth mentioning that the set $\mathbb{Y}_{00}$ plays the role of an initial allowance that is shrunk or expanded as a result of how close are closed-loop output signals to the prescribed region. In particular, $\mathbb{Y}_{00}$ can be selected a posteriori (after computation of the optimal trajectory) just for $t = 0$, so that the feasibility region of the algorithm is not affected by the introduction of average asymptotic constraints.

Asymptotic Average Performance

Since economic MPC does not necessarily lead to converging solutions, it is important to have bounds which estimate the asymptotic average performance of the closed-loop plant. To this end, the following dissipation inequality is needed for the approach with terminal penalty function:

$$V_f(f(x, \mathbb{K}(x))) \le V_f(x) - \ell(x, \mathbb{K}(x)) + \ell(x^*, u^*) \qquad (13)$$

which shall hold for all $x \in \mathbb{X}_f$. We are now ready to state the main bound on the asymptotic performance:

Theorem 3 *Let $x(0)$ be a feasible state for (8) and assume that either Assumption 1 or Assumption 2 together with (13) hold. Then, the closed-loop trajectory $x(t)$ resulting from receding horizon implementation of the feedback $u(t) = v^*(0)$ is well defined for all $t \in \mathbb{N}$ and fulfills*

$$\limsup_{T \to +\infty} \frac{\sum_{t=0}^{T-1} \ell(x(t), u(t))}{T} \le \ell(x^*, u^*). \quad (14)$$

The proof of this fact can be found in Angeli et al. (2012) and Amrit et al. (2011). When periodic solutions are known to outperform, in an average sense, the best equilibrium/control pair, one may replace terminal equality constraints by periodic terminal constraints (see Angeli et al. 2012). This leads to an asymptotic performance at least as good as that of the solution adopted as a terminal constraint.

Stability and Convergence

It is well known that the cost-to-go $V(x)$ as defined in (8) is a natural candidate Lyapunov function for the case of tracking MPC. In fact, the following estimate holds along solutions of the closed-loop system:

$$V(x(t+1)) \le V(x(t)) - \ell(x(t), u(t)) + \ell(x^*, u^*). \quad (15)$$

This shows, thanks to inequality (5), that $V(x(t))$ is nonincreasing. Owing to this, stability and convergence can be easily achieved under mild additional technical assumptions. While property (15) holds for economic MPC, both in the case of terminal equality constraint and terminal penalty function, it is no longer true that (5) holds. As a matter of fact, x^* might even fail to be an equilibrium of the closed-loop system, and hence, convergence and stability cannot be expected in general.

Intuitively, however, when the most profitable operating regime is an equilibrium, the average performance bound provided by Theorem 3 seems to indicate that some form of stability or convergence to x^* could be expected. This is true under an additional dissipativity assumption which is closely related to the property of optimal operation at steady state.

Definition 1 A system is strictly dissipative with respect to the supply function $s(x, u)$ if there exists a continuous function $\lambda : X \to \mathbb{R}$ and $\rho : X \to \mathbb{R}$ positive definite with respect to x^* such that for all x and u in $X \times U$, it holds:

$$\lambda(f(x, u)) \le \lambda(x) + s(x, u) - \rho(x). \quad (16)$$

The next result highlights the connection between dissipativity of the open-loop system and stability of closed-loop economic MPC.

Theorem 4 *Assume that either Assumption 1 or Assumption 2 together with (13) hold. Let the system (1) be strictly dissipative with respect to the supply function $s(x, u) = \ell(x, u) - \ell(x^*, u^*)$ as from Definition 1 and assume there exists a neighborhood of feasible initial states containing x^* in its interior. Then provided V is continuous at x^*, x^* is an asymptotically stable equilibrium with basin of attraction equal to the set of feasible initial states.*

See Angeli et al. (2012) and Amrit et al. (2011) for proofs and discussions. Convergence results are also possible for the case of economic MPC subject to average constraints. Details can be found in Müller et al. (2013a).

Hereby it is worth mentioning that finding a function satisfying (16) (should one exist) is in general a hard task (especially for nonlinear systems and/or nonconvex stage costs); it is akin to the problem of finding a Lyapunov function and therefore general construction methods do not exist. Let us emphasize, however, that while existence of a storage function λ is a sufficient condition to ensure convergence of closed-loop economic MPC, formulation and resolution of the optimization problem (8) can be performed irrespectively of any explicit knowledge of such function. Also, we point out that existence of λ as in Definition 1 and Theorem 4 is only possible if the optimal infinite-horizon regime of operation for the system is an equilibrium.

Summary and Future Directions

Economic model predictive control is a fairly recent and active area of research with great potential in those engineering applications where economic profitability is crucial rather than tracking performance.

The technical literature is rapidly growing in application areas such as chemical engineering

(see Heidarinejad 2012) or power systems engineering (see Hovgaard et al. 2010; Müller et al. 2013a) where system's output is in fact physical outflows which can be stored with relative ease.

We only dealt with the basic theoretical developments and would like to provide pointers to interesting recent and forthcoming developments in this field:

- Generalized terminal constraints: possibility of enlarging the set of feasible initial states by using arbitrary equilibria as terminal constraints, possibly to be updated on line in order to improve asymptotic performance (see Fagiano and Teel 2012; Müller et al. 2013b).
- Economic MPC without terminal constraints: removing the need for terminal constraints by taking a sufficiently long control horizon is an interesting possibility offered by standard tracking MPC. This is also possible for economic MPC at least under suitable technical assumptions as investigated in Grüne (2012, 2013).
- The basic developments presented in the previous paragraph only deal with systems unaffected by uncertainty. This is a severe limitation of current approaches and it is to be expected that, as for the case of tracking MPC, a great deal of research in this area could be developed in the future. In particular, both deterministic and stochastic uncertainties are of interest.

Cross-References

▶ Model-Predictive Control in Practice
▶ Optimization Algorithms for Model Predictive Control

Recommended Reading

Papers Amrit et al. (2011), Angeli et al. (2011, 2012), Diehl et al. (2011), Müller et al. (2013a), and Rawlings et al. (2008) set out the basic technical tools for performance and stability analysis of EMPC. To readers interested in the general theme of optimization of system's economic performance and its relationship with classical turnpike theory in economics, please refer to Rawlings and Amrit (2009). Potential applications of EMPC are described in Hovgaard et al. (2010), Heidarinejad (2012), and Ma et al. (2011) while Rawlings et al. (2012) is an up-to-date survey on the topic. Fagiano and Teel (2012) and Grüne (2012, 2013) deal with the issue of relaxation or elimination of terminal constraints, while Müller et al. (2013b) explore the possibility of adaptive terminal costs and generalized equality constraints.

Bibliography

Amrit R, Rawlings JB, Angeli D (2011) Economic optimization using model predictive control with a terminal cost. Annu Rev Control 35:178–186

Angeli D, Amrit R, Rawlings JB (2011) Enforcing convergence in nonlinear economic MPC. Paper presented at the 50th IEEE conference on decision and control and european control conference (CDC-ECC), Orlando 12–15 Dec 2011

Angeli D, Amrit R, Rawlings JB (2012) On average performance and stability of economic model predictive control. IEEE Trans Autom Control 57:1615–1626

Diehl M, Amrit R, Rawlings JB (2011) A Lyapunov function for economic optimizing model predictive control. IEEE Trans Autom Control 56:703–707

Fagiano L, Teel A (2012) Model predictive control with generalized terminal state constraint. Paper presented at the IFAC conference on nonlinear model predictive control, Noordwijkerhout 23–27 Aug 2012

Grüne L (2012) Economic MPC and the role of exponential turnpike properties. Oberwolfach Rep 12:678–681

Grüne L (2013) Economic receding horizon control without terminal constraints. Automatica 49:725–734

Heidarinejad M (2012) Economic model predictive control of nonlinear process systems using Lyapunov techniques. AIChe J 58:855–870

Hovgaard TG, Edlund K, Bagterp Jorgensen J (2010) The potential of Economic MPC for power management. Paper presented at the 49th IEEE conference on decision and control, Atlanta 15–17 Dec 2010

Ma J, Joe Qin S, Li B et al (2011) Economic model predictive control for building energy systems. Paper presented at the IEEE innovative smart grid technology conference, Anaheim, 17–19 Jan 2011

Müller MA, Angeli D, Allgöwer F (2013a) On convergence of averagely constrained economic MPC and necessity of dissipativity for optimal steady-state operation. Paper presented at the 2013 IEEE American control conference, Washington DC, 17–19 June 2013

Müller MA, Angeli D, Allgöwer F (2013b) Economic model predictive control with self-tuning terminal weight. Eur J Control 19:408–416

Rawlings JB, Amrit R (2009) Optimizing process economic performance using model predictive control. In: Magni L, Raimondo DM, Allgöwer F (eds) Nonlinear model predictive control. Lecture notes in control and information Sciences, vol 384. Springer, Berlin, pp 119–138

Rawlings JB, Bonne D, Jorgensen JB et al (2008) Unreachable setpoints in model predictive control. IEEE Trans Autom Control 53:2209–2215

Rawlings JB, Angeli D, Bates CN (2012) Fundamentals of economic model predictive control. Paper presented at the IEEE 51st annual conference on decision and control (CDC), Maui 10–13 Dec 2012

EKF

▶ Extended Kalman Filters

Electric Energy Transfer and Control via Power Electronics

Fred Wang
University of Tennessee, Knoxville, TN, USA

Abstract

Power electronics and their applications for electric energy transfer and control are introduced. The fundamentals of the power electronics are presented, including the commonly used semiconductor devices and power converter circuits. Different types of power electronic controllers for electric power generation, transmission and distribution, and consumption are described. The advantages of power electronics over traditional electromechanical or electromagnetic controllers are explained. The future directions for power electronic application in electric power systems are discussed.

Keywords

Electric energy control; Electric energy transfer; Power electronics

Introduction

Modern society runs on electricity or electric energy. The electric energy generally must be transferred before consumption since the energy sources, such as thermal power plants, hydro dams, and wind farms, are often some distances away from the loads. In addition, electric energy needs to be controlled as well since the energy transfer and use often require electricity in a form different from the raw form generated at the source. Examples are the voltage magnitude and frequency for long distance transmission; the voltage needs to be stepped up at the sending end to reduce the energy loss along the lines and then stepped down at the receiving end for users; for many modern consumer devices, DC voltage is needed and obtained through transforming the 50 or 60 Hz utility power. Note that electric energy transfer and control is often used interchangeably with the electric power transfer and control. This is because the modern electric power systems have very limited energy storage and the energy generated must be consumed at the same time.

Since the beginning of the electricity era, electric energy transfer and control technologies have been an essential part of electric power systems. Many types of equipment were invented and applied for these purposes. The commonly used equipment includes electric transmission and distribution lines, generators, transformers, switchgears, inductors or reactors, and capacitor banks. The traditional equipment has limited control capability. Many cannot be controlled at all or can only be connected or disconnected with mechanical switches, others with limited range, such as transformers with tap changers. Even with fully controllable equipment such as generators, the control dynamics is relatively slow due to the electromechanical or magnetic nature of the controller.

Power electronics are based on semiconductor devices. These devices are derivatives from transistors and diodes used in microelectronic circuits with the additional large power handling capability. Due to their electronic nature, power electronic devices are much more flexible and faster than their electromechanical or electromagnetic

counterparts for electric energy transfer and control. Since the advent of power electronics in the 1950s, they have steadily gained ground in power system applications. Today, power electronic controllers are an important part of equipment for electric energy transfer and control. Their roles are growing rapidly with the continuous improvement of the power electronic technologies.

Fundamentals of Power Electronics

Different from semiconductor devices in microelectronics, the power electronic devices only act as switches for desired control functions, such that they incur minimum losses when they are either on (closed) or off (open). As a result, the power electronic controllers are basically the switching circuits. The semiconductor switches are therefore the most important elements of the power electronic controllers. Since the 1950s, many different types of power semiconductor switches have been developed and can be selected based on the applications.

The performance of the power semiconductor devices is mainly characterized by their voltage and current ratings, conduction or on-state loss, as well as the switching speed (or switching frequency capability) and associated switching loss. Main types of power semiconductor devices are listed with their symbols and state-of-the-art rating and frequency range shown in Table 1:

- Power diode – a two terminal device with similar characteristics to diodes used in microelectronics but with higher-voltage and power ratings.

- Thyristor – also called SCR (silicon-controlled rectifier). Unlike diode, thyristor is a three-terminal device with an additional gate terminal. It can be turned on by a current pulse through gate but can only be turned off when the main current goes to zero with external means. Thyristor has low conduction loss but slow switching speed.
- GTO – stands for gate-turn-off thyristor. GTO can be turned on similarly as a regular thyristor and can also be turned off with a large negative gate current pulse. GTO has been largely replaced by IGBT and IGCT due to its complex gate driving needs and slow switching speed.
- Power BJT – similar to bipolar transistor for microelectronics and requires a sustained gate current to turn on and off. It has been replaced by IGBT and power MOSFET with simpler gate signals and faster switching speed.
- Power MOSFET – similar to metal-oxide semiconductor field effect transistor for microelectronics and can be turned on and off with a gate voltage signal. It is the fastest device available but has relatively high conduction loss and relatively low-voltage/power ratings.
- IGBT – stands for insulated-gate bipolar transistor. Unlike regular BJT, it can be turned on and off with a gate voltage like MOSFET. It has relatively low conduction loss and fast switching speed. IGBT is becoming the workhorse of the power electronics for high power applications.

Electric Energy Transfer and Control via Power Electronics, Table 1 Commonly use Si-based power semiconductor devices and their ratings

Types	Symbol	Voltage	Current	Switching frequency
Power diodes		Max 80 kV, typical < 10 kV	10 kA	Various
Thyristor		Max 8 kV	4.5 kA	AC line frequency
GTO		Max 10 kV	6.5 kA	<500 Hz
Power MOSFET		Max 4.5 kV, typical < 600 V	1.6 kA	10 s of kHz to MHz
IGBT		Max 6.5 kV, typical > 600 V	2.4 kA	1 kHz to 10 s of kHz
IGCT		Max 10 kV, typical > 4.5 kV	6.5 kA	<2 kHz

- IGCT – stands for integrated-gate-commutated thyristor. It is basically a GTO with an integrated gate drive circuit allowing a hard driven turnoff. It therefore has faster switching speed than regular GTO but slower than IGBT.

Except for diodes, all other devices above can be turned on and/or off through a gate signal, so they are active switches, while diodes are called passive switch.

With different types of power semiconductors, many power electronics circuits have been developed. Based on their functions, they can be classified as:

- Rectifier – rectifiers convert AC to DC. Depending on AC sources, rectifiers can be three phase or single phase; depending on device types, they can be passive (diode based), phase controlled (thyristor controlled), or actively switched.
- Inverter – inverters convert DC to AC. They again can be three phase or single phase. Inverters generally require active switching devices.
- DC-DC converter – also called choppers, DC-DC converters convert one DC voltage level to another. Sometimes they also contains a magnetic isolation. DC-DC converters can have unidirectional or bidirectional power flow and generally requires active switching devices.
- AC-AC converter – directly converts one AC to another, either only the voltage magnitude or both magnitude and frequency. The former can also be called AC switch, and the latter can

be called frequency changer. Active devices are needed for these types of converters.

There are a variety of converter topologies for each type of the converters listed above. The most commonly used basic topologies for power system applications are shown in Fig. 1. These basic topologies can be expanded through paralleling or series of devices and/or converters to achieve higher current and voltage ratings. Other variations such as multilevel converters are also popular for high-voltage applications using lower-voltage rating devices.

It should be noted that passive components, i.e., inductors and capacitors, are essential parts of power electronic converters. In fact, power electronic converters transfer or control the electric energy by storing it temporarily in inductors or capacitors while reformatting the original voltage or current waveform through switching actions. The other key function of the passives is filtering the harmonics caused by switching.

Power Electronic Controller Types for Energy Transfer and Control

For almost all traditional non-power-electronic equipment for electric energy transfer and control, there can be corresponding power electronic-based counterpart, often with better controllability. However, power electronic equipment can be more expensive and therefore only used when it provides better overall performance and cost benefits. In other cases, only power electronic equipment can achieve the required control functions.

a Thyristor based rectifier   **b** Voltage source inverter (VSI)   **c** Bi-directional AC switch
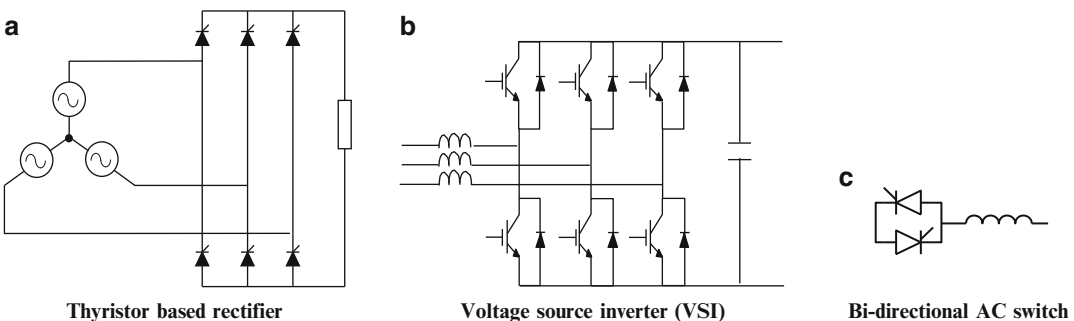
Electric Energy Transfer and Control via Power Electronics, Fig. 1 Commonly used basic power electronics converter topologies (only one phase shown for the AC switch)

The power electronic controllers can be categorized as for energy generation, delivery, and consumption. For generation, the thermal or hydro generators both use synchronous machines with excitation windings on the rotor, which require DC current. A thyristor-based rectifier, called exciter, is generally used for this purpose. Wind turbine generators usually use a back-to-back VSI to interface to the AC grid, and PV solar sources use a DC-DC converter cascaded with a VSI.

Power electronic controllers for transmission and distribution controllers include so-called flexible AC transmission systems (FACTS) and high-voltage DC transmission (HVDC). Some of the more commonly used controllers and their functions and circuit topologies are listed in Table 2.

The main power electronic controllers for loads include variable speed motor drives; electronic ballast for fluorescent lights and power supplies for LED; various power supplies for computer, IT, and other electronic loads; and chargers for electric vehicles. The percentage of power electronics controlled loads in power systems have been steadily increasing. Power electronics can generally result in improved performance and efficiency.
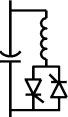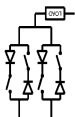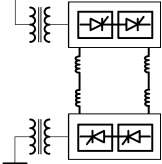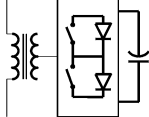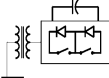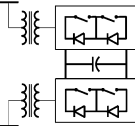
Future Directions

Power electronics have progressed steadily since the invention of thyristors in the 1950s. The progress is in all aspects, semiconductor devices, passives, circuits, control, and system integration, leading to converter systems with better performance, higher efficiency, higher power density, higher reliability, and lower cost. Because of these progresses, the power electronics applications in power systems have become more and more widespread. However, in general, power electronic controllers are still not sufficiently cost-effective, reliable, or efficient. Many improvements are needed and expected, especially in the following areas:

- Semiconductor devices – Devices used today are almost exclusively based on silicon. The emerging devices based on wide-bandgap materials such as SiC and GaN are expected to revolutionize power electronics with their capabilities of higher voltage, lower loss, faster switching speed, higher temperature, and smaller size.
- Power electronic converters – More cost-effective and reliable converters will be developed as a result of better devices, passive components, and circuit structures. Modular, distributed, and hybrid with non-power-electronics approaches are expected to result in overall better benefits.
- Enhanced functions – Power electronic controllers can be designed to have multiple functions in the system. For example, wind and PV solar inverters can provide reactive power to the grid in addition to transferring real energy. Today, power electronic controllers are mostly locally controlled. With better measurement and communication technologies, they may be controlled over a wide area for supporting the system level functions.
- New applications – The new applications for future power system include DC grid based on multiterminal HVDC and energy storage. Critical technologies include cost-effective and efficient DC transformers and DC circuit breakers. Power electronics will play key roles in these technologies.

Cross-References

▶ Cascading Network Failure in Power Grid Blackouts
▶ Coordination of Distributed Energy Resources for Provision of Ancillary Services: Architectures and Algorithms
▶ Lyapunov Methods in Power System Stability
▶ Power System Voltage Stability
▶ Small Signal Stability in Electric Power Systems
▶ Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency

E

Electric Energy Transfer and Control via Power Electronics, Table 2 Commonly used power electronic controllers for transmission and distribution

Controller	Oneline configuration	System functions	Control principle	Basic PE function
SVC – static VAR compensator with thyristor-controlled reactor and capacitor		• Stability enhancement • Voltage regulation and VAR compensation	VAR control through varying L and C in shunt connection	Controlled bidirectional AC switch
TCSC – thyristor-controlled series capacitor		• Power flow control • Stability enhancement • Fault current limiting	Power and VAR control through varying C and L in series connection	Controlled bidirectional AC switch
SSTS – solid state transfer switch		Power supply transfer for reliability and power quality	On and off control	Controlled bidirectional AC switch
HVDC (classic)		• System Interconnection • Power flow control • Stability enhancement	Power control through back-to-back converters in shunt connections	Bidirectional AC/DC current source converter
SSSC – static series synchronous compensator		• Power flow control • Stability enhancement	VAR control through voltage control in series connection	Bidirectional AC/DC voltage source converter
STATCOM – static synchronous compensator		• Stability enhancement • Voltage regulation & VAR compensation	VAR control through current control in shunt connection	Bidirectional AC/DC voltage source converter
HVDC (voltage source)		• System Interconnection • Power flow control • stability enhancement	Power and VAR control through back-to-back converters in shunt connections	Bidirectional AC/DC voltage source converter

Bibliography

Bayegen M (2001) A vision of the future grid. IEEE Power Eng Rev. vol 12, pp 10–12

Hingorani NG, Gyugyi L (2000) Understanding FACTS – concepts and technology of flexible AC transmission systems. IEEE Press, New York

Rahimo M, Klaka S (2009) High voltage semiconductor technologies. In: 13th European conference on power electronics and applications, EPE'09, Barcelona, pp 1–10

Wang F, Rosado S, Boroyevich D (2003) Open modular power electronics building blocks for utility power system controller applications. IEEE PESC, Acapulco, pp 1792–1797, 15–19 June 2003

Engine Control

Luigi del Re
Johannes Kepler Universität, Linz, Austria

Abstract

Engine control is the enabling technology for efficiency, performance, reliability, and cleanliness of modern vehicles for a wide variety of uses and users. It has also a paramount importance for many other engine applications like power plants. Engines are essentially chemical reactors, and the core task of engine control consists in preparing and starting the reaction (mixing the reactants and igniting the mixture) while the reaction itself is not controlled. The technical challenge derives from the combination of high complexity, wide range of conditions of use, performance requirements, significant time delays, and use of the constraints on the choice of components. In practice, engine control is to a large extent feed-forward control, feedback loops being used either for low-level control or for updating the feed-forward. Industrial engine control is based on very complex structures calibrated experimentally, but there is a growing interest for model-based control with stronger feedback action, supported by the breakthrough of new computational and communication possibilities, as well as the introduction of new sensors.

Introduction

Most vehicles are moved by internal combustion engines (ICE), whose key function is the conversion of chemical into mechanical energy, basically by oxidation, e.g., in the case of propane

$$C_3H_8 + 5O_2 = 3CO_2 + 4H_2O + 46.3 \text{ MJ/kg} \quad (1)$$
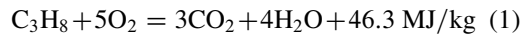
The chemical energy is first transformed into heat and then converted by the ICE into mechanical energy (Heywood 1988). The key task of engine control (Guzzella and Onder 2010; Kiencke and Nielssen 2005) is to make sure that the reactants (fuel and oxygen) meet in the right proportion ("mixture formation") and that the combustion is started (or "ignited") to deliver the required torque at the engine crankshaft. Several combustion processes are known, the most common ones being Otto and Diesel. For the first kind (also called SI for spark ignited), the mixture is prepared outside the combustion chamber and combustion is ignited by spark, while in the second one fuel is injected directly into the combustion chamber and combustion is ignited by compression (CI, compression ignited). GDI (gasoline direct injection) is a variant of SI engines with direct fuel injection as CI but spark ignition as SI.

Unfortunately, the chemical equation (1) is not the whole truth. Indeed, the way the mixture is prepared and ignited affects the efficiency of the conversion from thermal into mechanical energy, but also secondary reactions, like pollutant formation, and other aspects, like noise, vibrations and harshness (NVH), and mechanical fatigue and thus life expectancy. Furthermore, driveability requirements are primarily determined by the ability of an ICE to change fast its operating point, and this sets additional requirements to the engine control. These requirements have to

be met for all vehicles ín spite of production variability and under all relevant operating conditions, including all drivers, road, traffic, and weather conditions.

As first principle models are often not available or very time-consuming to tune and seldom precise enough, engine control is based on very complex heuristic descriptions which can be tuned experimentally and even automatically (Schoggl et al. 2002) – a modern engine control unit (ECU) can include up to 40.000 labels (parameters or maps). This structure is mainly feedforward, with feedback loops typically used for control of actuators, primarily calibrated under laboratory conditions but with adaptation loops designed to correct parameters to take in account production and wear effects. Figure 1 shows an engine test bench setup with the engine control unit (ECU) and a calibration system.

The Target System

Figure 2 shows the basic setup of an ICE as CI and SI. In both cases, the main components of an ICE are fuel path, air path, combustion chamber, and exhaust aftertreatment system.

Roughly speaking, ICEs exhibit three time scales. Changes in the setting of the fuel path – responsible to deliver the fuel to the combustion chamber – act very fast for CI and GDI engines (e.g., 50 Hz) and rather fast for SI engines (10 Hz or more). The same is not true for the air path which brings the gas mixture (fresh air and possibly recirculated exhaust gas) into the combustion chamber and is the slowest system (typically in the range of 0.5–2 Hz). In SI and GDI engines, spark timing can be changed for each combustion too. A still faster dynamics is associated with the combustion process itself, pressure sensors with the required dynamics to monitor it are being introduced in a growing number of applications, but until now no suitable actuators are available for its closed loop control. The torque demand changes typically with the vehicle dynamics, which are usually still slower than the air path.

The Control Tasks

The high-level control task can be defined as the minimization of the average fuel consumption while providing the required torque and respecting the constraints on emissions (i.e., nitrogen oxides and dioxides (NO_x) and particulate matter (PM)), noise, temperature, etc. The legislators in different countries have defined test procedure, including a specified road profile and corresponding emission limits. Figure 3 shows the progressive reduction of the limits and the speed profile used to assess this value.

Even if fuel consumption is not yet limited by law, the control problem associated can be stated as an optimal constrained control problem:

$$\min_{u(t)} \int_0^{1120} \dot{q}_f \, dt \qquad (2)$$

so that

$$v(t) = v_{\text{dem}}(t) \pm \Delta v \qquad (3)$$

and

$$\int_0^{1120} \dot{q}_i \, dt \leq Q_i \qquad (4)$$

where $u(t)$ are all available control inputs, 1120 is the duration of the European cycle, $v_{\text{dem}}(t)$ the corresponding speed, Δv the speed tolerance, $\dot{q}_f$ is the instantaneous fuel consumption, $\dot{q}_i$ each limited quantity (e.g., NOx), and Q_i the corresponding limit for the whole test. In practice, other criteria must be considered as well, like NVH, but even this problem is never solved using the standard tools of optimal control essentially for the nonlinearity (and following nonconvexity) of the problem, but even more for the lack of explicit models of sufficient quality relating the inputs to the target quantities, especially combustion depending quantities like emissions.

In practice, different simpler subproblems are solved separately and tuned to achieve sufficient results also in terms of the general problem to achieve the required performance. In the following, we concentrate on the main high-level tasks, omitting many others, e.g., all the control loops required for the correct operation of the single actuators.

E
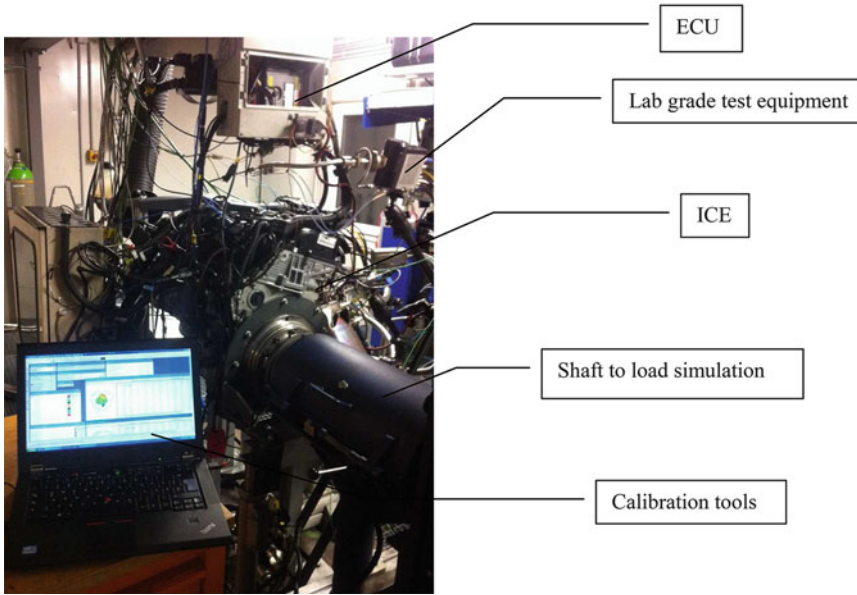
Engine Control, Fig. 1 Light duty engine test bench with ECU and calibration system
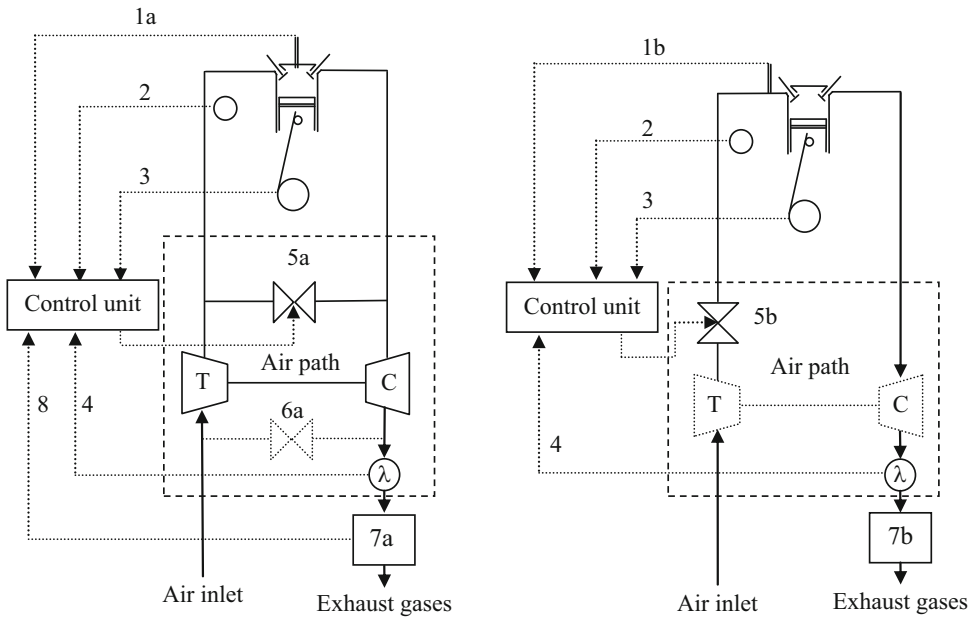
Engine Control, Fig. 2 Basic system scheme of CI (*left*) and SI (*right*) engines: *1a* Control of the injector opening, *1b* Injection premixing with air; *2* Measurement of the engine temperature; *3* Measurement of the engine rational speed; *4* Measurement of oxygen concentration in the exhaust gases; *5a* EGR valve, *5b* throttle valve, *6a* low-pressure EGR valve, *7a* Diesel exhaust after treatment (DOC, SCR, DPF), *7b* SI engine after treatment (3 way catalyst), *8* SCR dosing control
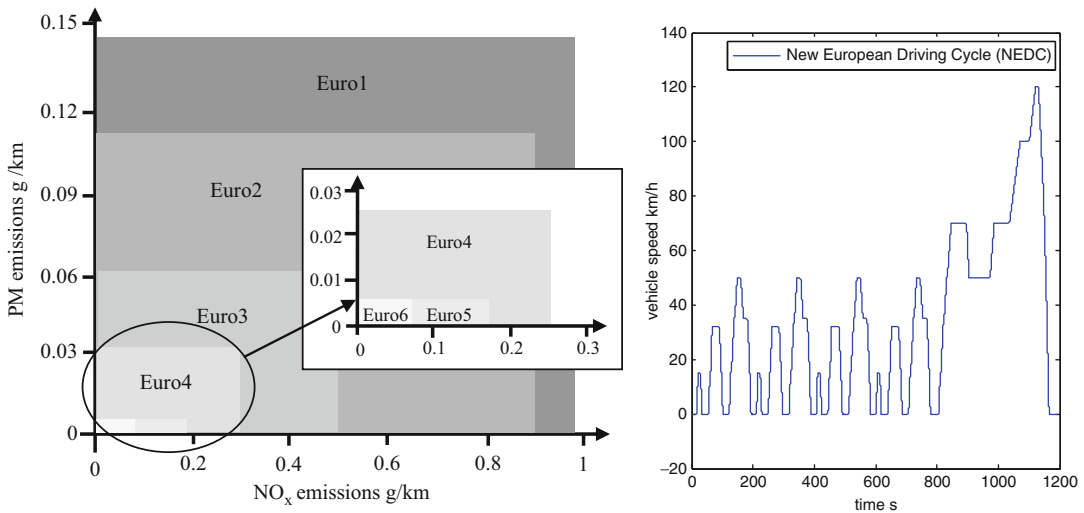
Engine Control, Fig. 3 *Left*: different steps of limits of emissions per km as defined by the European Union (Euro 1 introduced in 1991 and Euro 6 from 2014). *Right*: New European Driving Cycle (NEDC)

Air Path Control

The main source of oxygen for the reaction of Eq. (1) is ambient air which contains about 21 % oxygen. The engine – essentially a volumetric air pump – aspires air flow roughly proportional to the cylinder volume and the revolution speed of the engine. The amount of oxygen entering the combustion chamber, however, will depend also on temperature, pressure, and moisture. This flow can be reduced (as in the standard SI engines) by throttling, e.g., by adding an additional flow resistance between the air intake and the combustion chamber, or increased by compressing the fresh air, most commonly by turbocharging (especially in CI engines). A turbocharger consists essentially of a turbine, which transforms part of the enthalpy of the exhaust gas into mechanical power, and a compressor, driven by this power to compress the fresh air on its way to the combustion chamber, thus increasing both its density and temperature. Turbocharger operation is typically controlled either directly (for instance, with variable vane angles) or indirectly, by bypass valves which deviate the gas flows in parallel to the turbine.

If only ambient air is fed to the combustion chamber, a proportional amount of the other gases present in the atmosphere will enter the combustion chamber and be available for combustion side reactions as well. In the case of nitrogen, these reactions lead to the undesired formation of nitrogen oxides (NOx). Therefore, in some engines, especially in CI engines, part of the combusted gases are recirculated to the combustion chamber ("exhaust gas recirculation", EGR), providing advantages in terms of NOx reduction. While EGR is typically realized at high pressures (path HP in Fig. 1), it is realized also at low pressure (path LP), even though less frequently. Typically, the air path includes some coolers designed to increase gas densities.

Air path control is designed to track dynamical references, for instance, the total fresh air mass (MAF) entering the cylinder and the corresponding pressure (MAP), but also other quantities are possible. The references are typically generated by the calibration engineers on the basis of tests. The control inputs of the air path are mostly the turbine (and possibly compressor) steering angle, the EGR, and – if available – throttle(s) setpoints. Most commonly used sensors include a mass flow meter (hot film sensor), rather slow and dynamically not reliable, pressure, and temperature sensors, and sometimes the actual position of the valves is measured as well and the turbocharger speed.

Engine Control, Fig. 4
Pressure trace in a fired
cylinder of a CI engine
triggered by a pilot and a
main injection
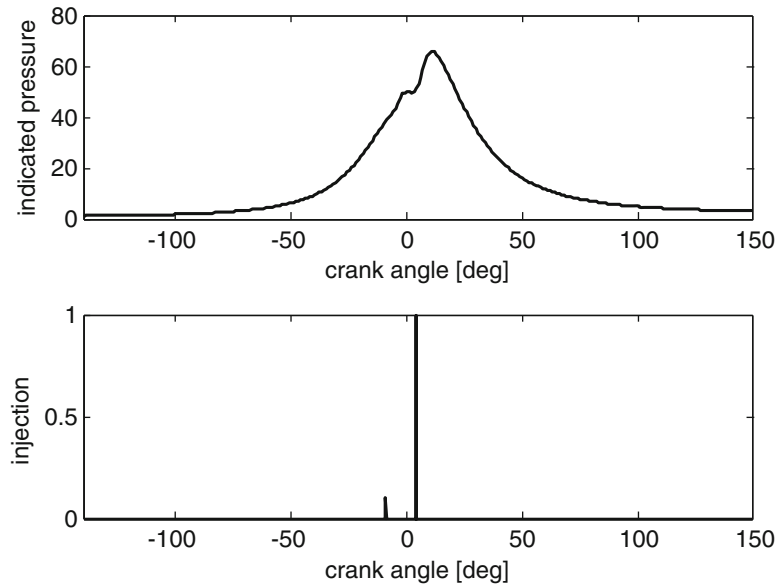
Fuel Path Control

The fuel path delivers the correct amount of fuel for the reaction (1). In almost every ICE, a rail is filled with fuel at a given pressure (from few bars for SI to about 2000 bars for CI), from which the required amount of fuel is injected into the cylinder. The injection can occur inside the combustion chamber (as for CI and GDI engines) or near to the intake valve ("port injection") for standard SI engines.

The injection amount is always set taking in account the available oxygen mass. In SI engines with three-way catalyst, the fuel injection is given by the stoichiometric condition. λ control uses an oxygen sensor in the exhaust to determine the actual fuel/oxygen ratio and if appropriate correct the injection tables. In CI and GDI the maximum fuel injection is limited to prevent smoke formation, typically by tables, even though λ control can be and is partly used (Amstutz and del Re 1995).

In SI engines with port injection, the liquid fuel is injected near to the inlet valve and is expected to vaporize due to the local temperature and pressure conditions. During load changes, however, it can happen that part of the fuel is not vaporized, remains on the duct wall ("wall wetting"), and vaporizes at a later time, leading in both cases to a deviation from the expected values (Turin et al. 1995),

which must be compensated by the injection control.

Injection in CI engines is typically splitted in a main injection for torque and a pilot injection for NVH control and sometimes also a post-injection for emission control or regeneration of aftertreatment devices. Figure 4 shows the typical effect of a pilot injection on the pressure trace of a CI engine.

Differences between injectors of different cylinders are compensated by cylinder balancing control (typically using irregularities in the engine acceleration). Rail pressure is also an important control variable for the direct injection.

Ignition

Once the combustion chamber is filled, the combustion can be started. In SI and GDI combustion is started by a spark) leading to a flame front which propagates through the whole combustion chamber. Very few SI engines have a second spark plug to better control the combustion. Under some circumstances, e.g., high temperature, an undesired auto-ignition ("knock") can occur with potentially catastrophic consequences for the engine durability but also unconventional NVH. To cope with this, SI engines have vibration sensors whose output is used to modify the engine operation, in particular the spark timing, to prevent it.

In CI engines, the injection leads almost immediately to the combustion which has more the character of an explosion and starts typically at several undefined locations.

Additional control during the combustion is up to now only theoretically feasible, as the combustion takes place in an extremely short time, but also because adequate actuators are not available.

Aftertreatment

As the combustion mixture will always contain more potential reactants than oxygen and fuel, side reactions will always take place, yielding toxic products, in particular NOx, incompletely burnt fuel (HC), carbon monoxide (CO), and particulate matter (PM). Even if much effort is spent on reducing their formation, this is almost never sufficient, so additional aftertreatment equipment is used. Table 1 gives an overview over the most common aftertreatment systems as well as over their control aspects.

Thermal Management

All main properties of engines are strongly affected by its temperature, which depends on the varying load conditions. Engine operation is typically optimal for a relatively narrow temperature range, the same is even more critical for the exhaust aftertreatment system. Engine heat is also required for other purposes (like defrosting of windshields in cold climates).

Thus the engine control system has two main tasks: bringing the engine and the exhaust aftertreatment system as fast as possible into the target temperature range and taking in account deviation from this target. The first task is performed both by control of the cooling circuit and by specific combustion-related measures, the second one by taking the measured or estimated temperature as input for the controllers.

Fast heating is especially important for SI engines, because almost all toxic emissions are produced when the three-way catalyst is cold. To achieve faster heating, SI engines tend to operate in a less fuel efficient, but "hotter" operation mode during this warm-up phase, one of the causes of increased consumption of cold engines and short trips.

Cranking Idle Speed and Gear Shifting Control

Initially, the engine is cranked by the starter until a relatively low speed and then injection starts bringing the engine to the minimum operational speed. If the injected fuel is not immediately burnt, very high emissions will arise. At cranking, the cylinder walls are typically very cold and combustion of a stoichiometric mixture is hardly possible. So engine control has the task to inject as little as possible but as much as needed only in the cylinder which is going to fire.

Normally an ICE is expected to provide a torque to the driveline, speed being the result of the balance between it and the load. In idle control, no torque is transmitted to the driveline, but the engine speed is expected to remain stable in spite of possible changes of local loads (like cabin climate control). This boils down to a robust control problem (Hrovat and Sun 1997).

Engine Control, Table 1   Main exhaust aftertreatment systems

System	Purpose	Control targets
Three-way catalyst	Reduction of HC, CO, and NOx by more than 98 %	Achieve fast and maintain operating temperature and keep $\lambda = 1$
Oxydation catalyst	Reduction of HC and CO, partly of PM	Achieve fast and maintain operating temperature and keep $\lambda > 1$
Particulate filter	Traps PM	Check trap state and regenerate by increasing exhaust temperature for short time if needed
NOx lean trap	Traps NOx	Estimate trap state and shift combustion to CO rich when required
Selective catalyst reaction	Reduces NOx	Estimate required quantity of additional reactant (urea) and dose it

Gear shifting requires several steps. Smoothness and speed of the shifting depend on the coordination of engine operating point change. Actual hardware developments (double clutches, automated gear boxes) make a better operation, but require precise control.

New Trends

The utilization environment of engine control is changing. On one side, customer and legislator expectations continue producing pressure, but there is a shift in priority from emissions to fuel efficiency and safety. Driver support systems, for instance, automated parking, are becoming the longer the more pervasive, and many functions must be included or affect immediately the ECU, even though they are frequently hosted on own control hardware. Hybrid vehicles are gaining popularity, and this implies a different operation mode for the engine, for instance, thermal management becomes much more complex for range extender vehicles with long "cold" phases.

Maybe even more important is the diffusion of new devices and communication possibilities, so that, for instance, fuel saving preview-based gear shifting can be easily implemented using infrastructure-to-vehicle information, or even just navigation data. Further extensions, like cooperative adaptive cruise control (CACC), plan to use vehicle-to-vehicle information to increase both safety and efficiency.

Against this background, there is a growing consciousness that the actual industrial approach based on huge calibration work is becoming the longer the less viable and bears a steadily increasing risk of wasting potential performance. Some model-based controls have already found their way into the ECU, and the academy has shown in several occasions that model-based control is able to achieve better performance, but it has not yet been shown how this could comply with other industrialization requirements.

Actually, new faster sensors (e.g., pressure sensors in the combustion chambers) are being introduced; the interest in model-based control (Alberer et al. 2012) and in system identification

techniques (del Re et al. 2010) are increasing, but they are not yet widespread.

Cross-References

▶ Powertrain Control for Hybrid-Electric and Electric Vehicles
▶ Transmission

Bibliography

Alberer D et al (2012) Identification for automotive systems. Springer, London

Amstutz A, del Re L (1995) EGO sensor based robust output control of EGR in diesel engines. IEEE Trans Control Syst Technol 3(1):39–48

del Re L et al (2010) Automotive model predictive control. Springer, Berlin

Guzzella L, Onder C (2010) Introduction to modeling and control of internal combustion engine systems, 2nd edn. Springer, Berlin

Heywood J (1988) Internal combustion engines fundamentals. McGrawHill, New York

Hrovat D, Sun J (1997) Models and control methodologies for IC engine idle speed control design. Control Eng Pract 5(8):1093–1100

Kiencke U, Nielssen L (2005) Automotive control systems: for engine, driveline, and vehicle, 2nd edn. Springer, New York

Schoggl P et al (2002) Automated EMS calibration using objective driveability assessment and computer aided optimization methods. SAE Trans 111(3):1401–1409

Turin RC, Geering HP (1995) Model-reference adaptive A/F-ratio control in an SI engine based on kalman-filtering techniques. In: Proceedings of the American Control Conference, Seattle, 1995, vol 6

Estimation and Control over Networks

Vijay Gupta
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Estimation and control of systems when data is being transmitted across nonideal communication channels has now become an

important research topic. While much progress has been made in the area over the last few years, many open problems still remain. This entry summarizes some results available for such systems and points out a few open research directions. Two popular channel models are considered – the analog erasure channel model and the digital noiseless model. Results are presented for both the multichannel and multisensor settings.

Keywords

Analog erasure channel; Digital noiseless channel; Networked control systems; Sensor fusion

Introduction

Networked control systems refer to systems in which estimation and control is done across communication channels. In other words, these systems feature data transmission among the various components – sensors, estimators, controllers, and actuators – across communication channels that may delay, erase, or otherwise corrupt the data. It has been known for a long time that the presence of communication channels has deep and subtle effects. As an instance, an asymptotically stable linear system may display chaotic behavior if the data transmitted from the sensor to the controller and the controller to the actuator is quantized. Accordingly, the impact of communication channels on the estimation/control performance and design of estimation/control algorithms to counter any performance loss due to such channels have both become areas of active research.

Preliminaries

It is not possible to provide a detailed overview of all the work in the area. This entry attempts to summarize the flavor of the results that are available today. We focus on two specific communication channel models – analog erasure channel and the digital noiseless channel. Although other

channel models, e.g., channels that introduce delays or additive noise, have been considered in the literature, these models are among the ones that have been studied the most. Moreover, the richness of the field can be illustrated by concentrating on these models.

An analog erasure channel model is defined as follows. At every time step k, the channel supports as its input a real vector $i(k) \in \mathbf{R}^t$ with a bounded dimension t. The output $o(k)$ of the channel is determined stochastically. The simplest model of the channel is when the output is determined by a Bernoulli process with probability p. In this case, the output is given by

$$o(k) = \begin{cases} i(k-1) & \text{with probability } 1-p \\ \phi & \text{otherwise,} \end{cases}$$

where the symbol ϕ denotes the fact that the receiver does not obtain any data at that time step and, importantly, recognizes that the channel has not transmitted any data. The probability p is termed the erasure probability of the channel. More intricate models in which the erasure process is governed by a Markov chain, or by a deterministic process, have also been proposed and analyzed. In our subsequent development, we will assume that the erasure process is governed by a Bernoulli process.

A digital noiseless channel model is defined as follows. At every time step k, the channel supports at its input one out of 2^m symbols. The output of the channel is equal to the input. The symbol that is transmitted may be generated arbitrarily; however, it is natural to consider the channel as supporting m bits at every time step and the specific symbol transmitted as being generated according to an appropriately design quantizer. Once again, additional complications such as delays introduced by the channel have been considered in the literature.

A general networked control problem consists of a process whose states are being measured by multiple sensors that transmit data to multiple controllers. The controllers generate control inputs that are applied by different actuators. All the data is transmitted across communication

channels. Design of control inputs when multiple controllers are present, even without the presence of communication channels, is known to be hard since the control inputs in this case have dual effect. It is, thus, not surprising that not many results are available for networked control systems with multiple controllers. We will thus concentrate on the case when only one controller and actuator is present. However, we will review the known results for the analog erasure channel and the digital noiseless channel models when (i) multiple sensors observe the same process and transmit information to the controller and (ii) the sensor transmits information to the controller over a network of communication channels with an arbitrary topology.

An important distinction in the networked control system literature is that of one-block versus two-block designs. Intuitively, the one-block design arises from viewing the communication channel as a perturbation to a control system designed without a channel. In this paradigm, the only block that needs to be designed is the receiver. Thus, for instance, if an analog erasure channel is present between the sensor and the estimator, the sensor continues to transmit the measurements as if no channel is present. However, the estimator present at the output of the channel is now designed to *compensate* for any imperfections introduced by the communication channel. On the other hand, in the two-block design paradigm, both the transmitter and the receiver are designed to optimize the estimation or control performance. Thus, if an analog erasure channel is present between the sensor and the estimator, the sensor can now transmit an appropriate function of the information it has access to. The transmitted quantity needs to satisfy the constraints introduced by the channel in terms of the dimensions, bit rate, power constraints, and so on. It is worth remembering that while the two-block design paradigm follows in spirit from communication theory where both the transmitter and the receiver are design blocks, the specific design of these blocks is usually much more involved than in communication theory. It is not surprising that in general performance with

two-block designs is better than the one-block designs.

Analog Erasure Channel Model

Consider the usual LQG formulation. A linear process of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k),$$

with state $x(k) \in \mathbf{R}^d$ and process noise $w(k)$ is controlled using a control input $u(k) \in \mathbf{R}^m$. The process noise is assumed to be white, Gaussian, zero mean, with covariance Σ_w. The initial condition $x(0)$ is also assumed to be Gaussian and zero mean with covariance Π_0. The process is observed by n sensors, with the i-th sensor generating measurements of the form

$$y_i(k) = C_i x(k) + v_i(k),$$

with the measurement noise $v_i(k)$ assumed to be white, Gaussian, zero mean, with covariance Σ_v^i. All the random variables in the system are assumed to be mutually independent. We consider two cases:

- If $n = 1$, the sensor communicates with the controller across a network consisting of multiple communication channels connected according to an arbitrary topology. Every communication channel is modeled as an analog erasure channel with possibly a different erasure probability. The erasure events on the channels are assumed to be independent of each other, for simplicity. The sensor and the controller then form two nodes of a network each edge of which represents a communication channel.
- If $n > 1$, then every sensor communicates with the controller across an individual communication channel that is modeled as an analog erasure channel with possibly a different erasure probability. The erasure events on the channels are assumed to be independent of each other, for simplicity.

The controller calculates the control input to optimize a quadratic cost function of the form

$$J_K = E\left[\sum_{k=0}^{K-1} \left(x^T(k)Qx(k) + u^T(k)Ru(k) \right) \right.$$

$$\left. + x^T(K)P_Kx(K) \right].$$

All the covariance matrices and the cost matrices Q, R, and P_K are assumed to be positive definite. The pair (A, B) is controllable and the pair (A, C) is observable, where C is formed by stacking the matrices C_i's. The system is said to be stabilizable if there exists a design (within the specified one-block or two-block design framework) such that the cost $\lim_{K\to\infty} \frac{1}{K} J_K$ is bounded.

A Network of Communication Channels

We begin with the case when $N = 1$ as mentioned above. The one-block design problem in the presence of a network of communication channels is identical to the one-block design as if only one channel were present. This is because the network can be replaced by an "equivalent" communication channel with the erasure probability as some function of the reliability of the network. This can lead to poor performance, since the reliability may decrease quickly as the network size increases. For this reason, we will concentrate on the two-block design paradigm.

The two-block design paradigm permits the nodes of the network to process the data prior to transmission and hence achieve much better performance. The only constraint imposed on the transmitter is that the quantity that is transmitted is a causal function of the information that the node has access to, with a bounded dimension. The design problem can be solved using the following steps. The first step is to prove that a separation principle holds if the controller knows the control input applied by the actuator at every time step. This can be the case if the controller transmits the control input to the actuator across a perfect channel or if the control input is transmitted across an analog erasure channel but the actuator can transmit an acknowledgment to the controller. For simplicity, we assume that the controller transmits the control input to the actuator across a perfect channel. The separation principle states that the optimal performance is achieved if the control input is calculated using the usual LQR control law, but the process state is replaced by the minimum mean squared error (MMSE) estimate of the state. Thus, the two-block design problem needs to be solved now for an optimal *estimation* problem.

The next step is to realize that for any allowed two-block design, an upper bound on estimation performance is provided by the strategy of every node transmitting every measurement it has access to at each time step. Notice that this strategy is not in the set of allowed two-block designs since the dimension of the transmitted quantity is not bounded with time. However, the same estimate is calculated at the decoder if the sensor transmits an estimate of the state at every time step and every other node (including the decoder) transmits the latest estimate it has access to from either its neighbors or its memory. This algorithm is recursive and involves every node transmitting a quantity with bounded dimension, however, since it leads to calculation of the same estimate at the decoder, and is, thus, optimal. It is worth remarking that the intermediate nodes do not require access to the control inputs. This is because the estimate at the decoder is a linear function of the control inputs and the measurements: thus, the effect of control inputs in the estimate can be separated from the effect of the measurements and included at the controller. Moreover, as long as the closed loop system is stable, the quantities transmitted by various nodes are also bounded. Thus, the two-block design problem can be solved.

The stability and performance analysis with the optimal design can also be performed. As an example, a necessary and sufficient stabilizability condition is that the inequality

$$p_{\text{maxcut}}\rho(A)^2 < 1,$$

holds, where $\rho(A)$ is the spectral radius of A and p_{maxcut} is the max-cut probability evaluated as follows. Generate cut-sets from the network by dividing the nodes into two sets – a source

set containing the sensor and a sink set containing the controller. For each cut-set, obtain the cut-set probability by multiplying the erasure probabilities of the channels from the source set to the sink set. The max-cut probability is the maximum such cut-set probability. The necessity of the condition follows by recognizing that the channels from the source set to the sink set need to transmit data at a high enough rate even if the channels within each set are assumed not to erase any data. The sufficiency of the condition follows by using the Ford-Fulkerson algorithm to reduce the network into a collection of parallel paths from the sensor to the controller such that each path has links with equal erasure probability and the product of these probabilities for all paths is the max-cut probability. More details can be found in Gupta et al. (2009a).

Multiple Sensors

Let us now consider the case when the process is observed using multiple sensors that transmit data to a controller across an individual analog erasure channel. A separation principle to reduce the control design problem into the combination of an LQR control law and an estimation problem can once again be proven. Thus, the two-block design for the estimation problem asks the following question: what quantity should the sensors transmit such that the decoder is able to generate the optimal MMSE estimate of the state at every time step, given all the information the decoder has received till that time step. This problem is similar to the track-to-track fusion problem that has been studied since the 1980s and is still open for general cases (Chang et al. 1997). Suppose that at time k, the last successful transmission from sensor i happened at time $k_i \leq k$. The optimal estimate that the decoder can ever hope to achieve is the estimate of the state $x(k)$ based on all measurements from the sensor 1 till time k_1, from sensor 2 till time k_2, and so on. However, it is not known whether this estimate is achievable if the sensors are constrained to transmit real vectors with a bounded dimension. A fairly intuitive encoding scheme is if the sensors transmit the *local* estimates of the state based on their own measurements. However,

it is known that the *global* estimate cannot, in general, be obtained from local estimates because of the correlation introduced by the process noise. If erasure probabilities are zero, or if the process noise is not present, then the optimal encoding schemes are known. Another case for which the optimal encoding schemes are known is when the estimator sends back acknowledgments to the encoders.

Transmitting local estimates does, however, achieve optimal stability conditions as compared to the conditions obtained from the optimal (unknown) two-block design (Gupta et al. 2009b). As an example, the necessary and sufficient stability conditions for the two sensor cases are given by

$$p_1 \rho(A_1)^2 < 1$$
$$p_2 \rho(A_2)^2 < 1$$
$$p_1 p_2 \rho(A_3)^2 < 1,$$

where p_1 and p_2 are erasure probabilities from sensors 1 and 2, respectively, $\rho(A_1)$ is the spectral radius of the unobservable part of the matrix A from the second sensor, $\rho(A_2)$ is the spectral radius of the unobservable part of the matrix A from the first sensor, and $\rho(A_3)$ is the spectral radius of the observable part of the matrix A from both the sensors. The conditions are fairly intuitive. For instance, the first condition provides a bound on the rate of increase of modes for which only sensor 1 can provide information to the controller, in terms of how reliable the communication channel from the sensor 1 is.

Digital Noiseless Channels

Similar results as above can be derived for the digital noiseless channel model. For the digital noiseless channel model, it is easier to consider the system without either measurement or process noises (although results with such noises are available). Moreover, since quantization is inherently highly nonlinear, results such as separation between estimation and control are not available. Thus, encoders and controllers that optimize a

cost function such as a quadratic performance metric are not available even for the single sensor or channel case. Most available results thus discuss stabilizability conditions for a given data rate that the channels can support.

While early works used the one-block design framework to model the digital noiseless channel as introducing an additive white quantization noise, that framework obscures several crucial features of the channel. For instance, such an additive noise model suggests that at any bit rate, the process can be stabilized by a suitable controller. However, a simple argument can show that is not true. Consider a scalar process in which at time k, the controller knows that the state is within a set of length $l(k)$. Then, stabilization is possible only if $l(k)$ remains bounded as $k \to \infty$. Now, the evolution of $l(k)$ is governed by two processes: at every time step, this uncertainty can be (i) decreased by a factor of at most 2^m due to the data transmission across the channel and (ii) increased by a factor of a (where a is the process matrix governing the evolution of the state) due to the process evolution. This implies that for stabilization to be possible, the inequality $m \geq \log_2(a)$ must hold. Thus, the additive noise model is inherently wrong. Most results in the literature formalize this basic intuition above (Nair et al. 2007).

A Network of Communication Channels
For the case when there is only one sensor that transmits information to the controller across a network of communication channels connected in arbitrary topology, an analysis similar to that done for analog erasure channels can be performed (Tatikonda 2003). A max-flow min-cut like theorem again holds. The stability condition now becomes that for any cut-set

$$\sum R_j > \sum_{\text{all unstable eigenvalues}} \log_2(\lambda_i),$$

where $\sum R_j$ is the sum of data rates supported by the channels joining the source set to sink set for any cut-set and λ_i are the eigenvalues of the process matrix A. Note that the summation on the right hand side is only over the unstable eigenvalues, since no information needs to be transmitted about the modes that are stable in open loop.

Multiple Sensors
The case when multiple sensors transmit information across an individual digital noiseless channel to a controller can also be considered. For every sensor i, define a rate vector $\{R_{i_1}, R_{i_2}, \cdots, R_{i_d}\}$ corresponding to the d modes of the system. If a mode j cannot be observed from the sensor i, set $R_{i_j} = 0$. For stability, the condition

$$\sum_i R_{ij} \geq \max(0, \lambda_j),$$

for every mode j must be satisfied. All such rate vectors stabilize the system.

Summary and Future Directions

This entry provided a brief overview of some results available in the field of networked control systems. Although the area is seeing intense research activity, many problems remain open. For control across analog erasure channels, most existing results break down if a separation principle cannot be proved. Thus, for example, if control packets are also transmitted to the actuator across an analog erasure channel, the LQG optimal two-block design is unknown. There is some recent work on analyzing the stabilizability under such conditions (Gupta and Martins 2010), but the problem remains open in general. For digital noiseless channels, controllers that optimize some performance metric are largely unknown. Considering more general channel models is also an important research direction (Martins and Dahleh 2008; Sahai and Mitter 2006).

Cross-References

▶ Averaging Algorithms and Consensus
▶ Oscillator Synchronization

Bibliography

Chang K-C, Saha RK, Bar-Shalom Y (1997) On optimal track-to-track fusion. IEEE Trans Aerosp Electron Syst AES-33:1271–1276

Gupta V, Martins NC (2010) On stability in the presence of analog erasure channels between controller and actuator. IEEE Trans Autom Control 55(1): 175–179

Gupta V, Dana AF, Hespanha J, Murray RM, Hassibi B (2009a) Data transmission over networks for estimation and control. IEEE Trans Autom Control 54(8):1807–1819

Gupta V, Martins NC, Baras JS (2009b) Stabilization over erasure channels using multiple sensors. IEEE Trans Autom Control 54(7):1463–1476

Martins NC, Dahleh M (2008) Feedback control in the presence of noisy channels: 'bode-like' fundamental limitations of performance. IEEE Trans Autom Control 53(7):1604–1615

Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. Proc IEEE 95(1):108–137

Sahai A, Mitter S (2006) The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link–Part I: scalar systems. IEEE Trans Inf Theory 52(8):3369–3395

Tatikonda S (2003) Some scaling properties of large distributed control systems. In: 42th IEEE conference on decision and control, Maui, Dec 2003

Estimation for Random Sets

Ronald Mahler
Eagan, MN, USA

Abstract

The random set (RS) concept generalizes that of a random vector. It permits the mathematical modeling of random systems that can be interpreted as random patterns. Algorithms based on RSs have been extensively employed in image processing. More recently, they have found application in multitarget detection and tracking and in the modeling and processing of human-mediated information sources. The purpose of this entry is to briefly summarize the concepts, theory, and practical application of RSs.

Keywords

Introduction

In ordinary signal processing, one models physical phenomena as "sources," which generate "signals" obscured by random "noise." The sources are to be extracted from the noise using optimal-estimation algorithms. Random set (RS) theory was devised about 40 years ago by mathematicians who also wanted to construct optimal-estimation algorithms. The "signals" and "noise" that they had in mind, however, were *geometric patterns in images*. The resulting theory, *stochastic geometry*, is the basis of the "morphological operators" commonly employed today in image-processing applications. It is also the basis for the theory of RSs. An important special case of RS theory, the theory of *random finite sets* (RFSs), addresses problems in which the patterns of interest consist of a finite number of points. It is the theoretical basis of many modern medical and other image-processing algorithms. In recent years, RFS theory has found application to the problem of detecting, localizing, and tracking unknown numbers of unknown, evasive point targets. Most recently and perhaps most surprisingly, RS theory provides a theoretically rigorous way of addressing "signals" that are *human-mediated*, such as natural-language statements and inference rules. The breadth of RS theory is suggested in the various chapters of Goutsias et al. (1997).

The purpose of this entry is to summarize the RS and RFS theories and their applications. It is divided in to the following sections: A Simple Example, Mathematics of Random Sets, Random Sets and Image Processing, Random Sets and Multitarget Processing, Random Sets and Human-Mediated Data, Summary and Future Directions, Cross-References, and Recommended Reading.

A Simple Example

To illustrate the concept of a RS, let us begin by examining a simple example: *locating stars in the nighttime sky*. We will proceed in successively more illustrative steps:

Locating a single non-dim star (estimating a random point). When we try to locate a star, we are trying to estimate its actual position – its "state" $\mathbf{x} = (\alpha_0, \theta_0)$ – in terms of its azimuth angle α_0 and elevation angle θ_0. When the star is dim but not too dim, its apparent position will vary slightly. We can estimate its position by averaging many measurements – i.e., by applying a *point estimator*.

Locating a very dim star (estimating an RS with at most one element). Assume that the star is so dim that, when we see it, it might be just a momentary visual illusion. Before we can estimate its position, we must *first estimate whether or not it exists*. We must record not only its apparent position $\mathbf{z} = (\alpha, \theta)$ (if we see it) but its *apparent existence* ε, with $\varepsilon = 1$ (we saw it) or $\varepsilon = 0$ (we did not). Averaging ε over many observations, we get a number q between 0 and 1. If $q > \frac{1}{4}$ (say), we could declare that the star probably actually is a star; and then we could average the non-null observations to estimate its position.

Locating multiple stars (estimating an RFS). Suppose that we are trying to locate *all* of the stars in some patch of sky. In some cases, two dim stars may be so close that they are difficult to distinguish. We will then collect three kinds of measurements from them: $Z = \emptyset$ (did not see either star), $Z = \{(\alpha, \theta)\}$ (we saw one or the other), or $Z = \{(\alpha_1, \theta_1), (\alpha_2, \theta_2)\}$ (saw both). The total collected measurement in the patch of sky is a finite set $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ of point measurements with $\mathbf{z}_j = (\theta_j, \alpha_j)$, where each $\mathbf{z}_i$ is random, where m is random, and where $m = 0$ corresponds to the null measurement $Z = \emptyset$.

Locating multiple stars in a quantized sky (estimation using imprecise measurements). Suppose that, for computational reasons, the patch of sky must be quantized into a finite number of hexagonal-shaped cells, $c_1, \ldots, c_M$. Then, the measurement from any star is not a specific point $\mathbf{z}$, but instead the cell c that contains $\mathbf{z}$. The measurement c is *imprecise* – a randomly varying hexagonal cell c. There are two ways of thinking about the total measurement collection. First, it is a finite set $Z = \{c_1', \ldots, c_m'\} \subseteq \{c_1, \ldots, c_M\}$ of cells. Second, it is the union $Z = c_1' \cup \ldots \cup c_m'$ of all of the observed cells – i.e., it is a *geometrical pattern*.

Locating multiple stars over an extended period of time (estimating multiple moving targets). As the night progresses, we must continually redetermine the existence and positions of each star – a process called *multitarget tracking*. We must also account for appearances and disappearances of the stars in the patch – i.e., for *target death and birth*.

Mathematics of Random Sets

The purpose of this section is to sketch the elements of the theory of random sets. It is organized as follows: General Theory of Random Sets, Random Finite Sets (Random Point Processes), and Stochastic Geometry. Of necessity, the material is less elementary than in later sections.

General Theory of Random Sets

Let $\mathfrak{Y}$ be a topological space – for example, an N-dimensional Euclidean space $\mathbb{R}^N$. The *power set* $2^{\mathfrak{Y}}$ of $\mathfrak{Y}$ is the class of all possible subsets $S \subseteq \mathfrak{Y}$. Any subclass of $2^{\mathfrak{Y}}$ is called a "hyperspace." The "elements" or "points" of a hyperspace are thus actually subsets of some other space. For a hyperspace to be of interest, one must extend the topology on $\mathfrak{Y}$ to it. There are many possible topologies for hyperspaces (Michael 1950). The most well studied is the *Fell-Matheron topology*, also called the "hit-and-miss" topology (Matheron 1975). It is applicable when $\mathfrak{Y}$ is Hausdorff, locally compact, and completely separable. It topologizes only the hyperspace $\mathfrak{c}(2^{\mathfrak{Y}})$ of all *closed* subsets C of $\mathfrak{Y}$. In this case, a *random* (closed) *set* Θ is a measurable mapping from some probability space into $\mathfrak{c}(2^{\mathfrak{Y}})$.

The Fell-Matheron topology's major strength is its relative simplicity. Let "$\Pr(\mathcal{E})$" denote the probability of a probabilistic event $\mathcal{E}$. Then, normally, the probability law of Θ would be

described by a very abstract probability measure $p_\Theta(O) = \Pr(\Theta \in O)$. This measure must be defined on the Borel-measurable subsets $O \subseteq \mathfrak{c}(2^{\mathfrak{Y}})$, with respect to the Fell-Matheron topology, where O is itself a class of subsets of $\mathfrak{Y}$. However, define the *Choquet capacity functional* by $c_\Theta(G) = \Pr(\Theta \cap G \neq \emptyset)$ for all open subsets $G \subseteq \mathfrak{Y}$. Then, the *Choquet-Matheron theorem* states that the probability law of Θ is completely described by the simpler, albeit nonadditive, measure $c_\Theta(G)$.

The theory of random sets has evolved into a substantial subgenre of statistical theory (Molchanov 2005). For estimation theory, the concept of the *expected value* $\mathbb{E}[\Theta]$ of a random set Θ is of particular interest. Most definitions of $\mathbb{E}[\Theta]$ are very abstract (Molchanov 2005, Chap. 2). In certain circumstances, however, more conventional-looking definitions are possible. Suppose that $\mathfrak{Y}$ is a Euclidean space and that $\mathfrak{c}(2^{\mathfrak{Y}})$ is restricted to $\mathfrak{K}(2^{\mathfrak{Y}})$, the *bounded, convex, closed subsets* of $\mathfrak{Y}$. If C, C' are two such subsets, their *Minkowski sum* is $C + C' = \{c + c' \mid c \in C, c' \in C'\}$. Endowed with this definition of addition, $\mathfrak{K}(2^{\mathfrak{Y}})$ can be homeomorphically and homomorphically embedded into a certain space of functions (Molchanov 2005, pp. 199–200). Denote this embedding by $C \longmapsto \phi_C$. Then, the expected value $\mathbb{E}[\Theta]$ of Θ, defined in terms of Minkowski addition, corresponds to the *conventional* expected value $\mathbb{E}[\phi_\Theta]$ of the random function ϕ_Θ.

Random Finite Sets (Random Point Processes)

Suppose that the $\mathfrak{c}(2^{\mathfrak{Y}})$ is restricted to $\mathfrak{f}(2^{\mathfrak{Y}})$, the class of *finite* subsets of $\mathfrak{Y}$. (In many formulations, $\mathfrak{f}(2^{\mathfrak{Y}})$ is taken to be the class of *locally finite* subsets of $\mathfrak{Y}$ – i.e., those whose intersection with compact subsets is finite.) A *random finite set* (RFS) is a measurable mapping from a probability space into $\mathfrak{f}(2^{\mathfrak{Y}})$. An example: the field of twinkling stars in some patch of a night sky. RFS theory is a particular mathematical formulation of *point process theory* (Daley and Vere-Jones 1998; Snyder and Miller 1991; Stoyan et al. 1995).

A *Poisson RFS* Ψ is perhaps the simplest nontrivial example of a random point pattern. It is specified by a *spatial distribution* $s(\mathbf{y})$ and an *intensity* μ. At any given instant, the probability that there will be n points in the pattern is $p(n) = e^{-\mu} \mu^n / n!$ (the value of the Poisson distribution). The probability that one of these n points will be $\mathbf{y}$ is $s(\mathbf{y})$. The function $D_\Psi(\mathbf{y}) = \mu \cdot s(\mathbf{y})$ is called the *intensity function* of Ψ.

At any moment, the point pattern produced by Ψ is a finite set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ of points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ in $\mathfrak{Y}$, where $n = 0, 1, \ldots$ and where $Y = \emptyset$ if $n = 0$. If $n = 0$ then Y represents the hypothesis that no objects at all are present. If $n = 1$ then $Y = \{\mathbf{y}_1\}$ represents the hypothesis that a single object $\mathbf{y}_1$ is present. If $n = 2$ then $Y = \{\mathbf{y}_1, \mathbf{y}_2\}$ represents the hypothesis that there are two distinct objects $\mathbf{y}_1 \neq \mathbf{y}_2$. And so on.

The *probability distribution* of Ψ – i.e., the probability that Ψ will have $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ as an instantiation – is entirely determined by its intensity function $D_\Psi(\mathbf{y})$:

$$\begin{aligned} f_\Psi(Y) &= f_\Psi(\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}) \\ &= e^{-mu} \cdot D_\Psi(\mathbf{y}_1) \cdots D_\Psi(\mathbf{y}_n) \end{aligned}$$

Every suitably well-behaved RFS Ψ has a probability distribution $f_\Psi(Y)$ and an intensity function $D_\Psi(\mathbf{y})$ (a.k.a. *first-moment density*). A Poisson RFS is unique in that $f_\Psi(Y)$ is completely determined by $D_\Psi(\mathbf{y})$.

Conventional signal processing is often concerned with single-object random systems that have the form

$$\mathbf{Z} = \eta(\mathbf{x}) + \mathbf{V}$$

where $\mathbf{x}$ is the state of the system; $\eta(\mathbf{x})$ is the "signal" generated by the system; the zero-mean random vector $\mathbf{V}$ is the random "noise" associated the sensor; and $\mathbf{Z}$ is the random measurement that is observed. The purpose of signal processing is to construct an estimate $\hat{\mathbf{x}}(\mathbf{z}_1, \ldots, \mathbf{z}_k)$ of $\mathbf{x}$, using the information contained in one or more draws $\mathbf{z}_1, \ldots, \mathbf{z}_k$ from the random variable $\mathbf{Z}$.

RFS theory is analogously concerned with random systems that have the form

$$\Sigma = \Upsilon(X) \cup \Omega$$

where a random finite point pattern $\Upsilon(X)$ is the "signal" generated by the point pattern X (which is an instantiation of a random point pattern Ξ); Ω is a random finite point "noise" pattern; Σ is the total random finite point pattern that has been observed; and "$\cup$" denotes set-theoretic union. One goal of RFS theory is to devise algorithms that can construct an estimate $\hat{X}(Z_1, \ldots, Z_k)$ of X, using multiple point patterns $Z_1, \ldots, Z_k \subseteq \mathfrak{Y}$ drawn from Σ. One approximate approach is that of estimating only the first-moment density $D_\Xi(\mathbf{x})$ of Ξ.

Stochastic Geometry

Stochastic geometry addresses more complicated random patterns. An example: the field of twinkling stars in a *quantized* patch of the night sky, in which case the measurement is the union $c_1 \cup \ldots \cup c_m$ of a finite number of hexagonally shaped cells.

This is one instance of a *germ-grain process* (Stoyan et al. 1995, pp. 59–64). Such a process is specified by two items: an RFS Ψ and a function $c_\mathbf{y}$ that associates with each $\mathbf{y}$ in $\mathfrak{Y}$ a closed subset $c_\mathbf{y} \subseteq \mathfrak{Z}$. For example, if $\mathfrak{Y} = \mathbb{R}^2$ is the real-valued plane, then $c_\mathbf{y}$ could be the disk of radius r centered at $\mathbf{y} = (x, y)$. Let $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ be a particular random draw from Ψ. The points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are the "germs," and $c_{\mathbf{y}_1}, \ldots, c_{\mathbf{y}_n}$ are the "grains" of this random draw from the germ-grain process Θ. The total pattern in $\mathfrak{Y}$ is the union $c_{\mathbf{y}_1} \cup \ldots \cup c_{\mathbf{y}_n}$ of the grains – a random draw from Θ. Germ-grain processes can be used to model many kinds of natural processes. One example is the distribution of graphite particles in a two-dimensional section of a piece of iron, in which case the $c_\mathbf{y}$ could be chosen to be line segments rather than disks.

Stochastic geometry is concerned with random binary images that have observation structures such as

$$\Theta = (S \cap \Delta) \cup \Omega$$

where S is a "signal" pattern; Δ is a random pattern that models obscurations; Ω is a random pattern that models clutter; and Θ is the total pattern that has been observed. A common simplifying assumption is that Ω and Δ^c are germ-grain processes. One goal of stochastic geometry is to devise algorithms that can construct an optimal estimate $\hat{S}(T_1, \ldots, T_k)$ of S, using multiple patterns $T_1, \ldots, T_k \subseteq \mathfrak{Y}$ drawn from Θ.

Random Sets and Image Processing

Both point process theory and stochastic geometry have found extensive application to image-processing applications. These are considered briefly in turn.

Stochastic Geometry and Image Processing. Stochastic geometry methods are based on the use of a "structuring element" B (a geometrical shape, such as a disk, sphere, or more complex structure) to modify an image.

The *dilation* of a set S by B is $S \oplus B$ where "$\oplus$" is Minkowski addition (Stoyan et al. 1995). Dilation tends to fill in cavities and fissures in images. The *erosion* of S is $S \ominus B = (S^c \oplus B^c)^c$ where "^{c}" indicates set-theoretic complement. Erosion tends to create and increase the size of cavities and fissures. *Morphological filters* are constructed from various combinations of dilation and erosion operators.

Suppose that a binary image $\Sigma = S$ has been degraded by some measurement process – for example, the process $\Theta = (S \cap \Delta) \cup \Omega$. Then, *image restoration* refers to the construction of an estimate $\hat{S}(T)$ of the original image S from a single degraded image $\Theta = T$. The restoration operator $\hat{S}(T)$ is *optimal* if it can be shown to be optimally close to S, given some concept of closeness. The *symmetric difference*

$$T_1 \sqcup T_2 = (T_1 \cup T_2) - (T_1 \cap T_2)$$

is a commonly used method for measuring the dissimilarity of binary images. It can be used to construct measures of distance between random images. One such distance is

$$d(\Theta_1, \Theta_2) = \mathbb{E}\left[|\Theta_1 \sqcup \Theta_2|\right]$$

where $|S|$ denotes the size of the set S and $\mathbb{E}[A]$ is the expected value of the random number A. Other distances require some definition of the expected value $\mathbb{E}[\Theta]$ of a random set Θ. It has been shown that, under certain circumstances, certain morphological operators can be viewed as consistent maximum a posteriori (MAP) estimators of S (Goutsias et al. 1997, p. 97).

RFS Theory and Image Processing. Positron-emission tomography (PET) is one example of the application of RFS theory. In PET, tissues of interest are suffused with a positron-emitting radioactive isotope. When a positron annihilates an electron in a suitable fashion, two photons are emitted in opposite directions. These photons are detected by sensors in a ring surrounding the radiating tissue. The location of the annihilation on the line can be estimated by calculating time difference of arrival.

Because of the physics of radioactive decay, the annihilations can be accurately modeled as a Poisson RFS Ψ. Since a Poisson RFS is completely determined by its intensity function $D_\Psi(\mathbf{x})$, it is natural to try to estimate $D_\Psi(\mathbf{x})$. This yields the spatial distribution $s_\Psi(\mathbf{y})$ of annihilations – which, in turn, is the basis of the PET image (Snyder and Miller 1991, pp. 115–119).

Random Sets and Multitarget Processing

The purpose of this section is to summarize the application of RFS theory to multitarget detection, tracking, and localization. An example: tracking the positions of stars in the night sky over an extended period of time.

Suppose that at time t_k there are an unknown number n of targets with unknown states $\mathbf{x}_1, \ldots, \mathbf{x}_n$. The state of the entire multitarget system is a finite set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $n \geq 0$. When interrogating a scene, many sensors (such as radars) produce a measurement of the form $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ – i.e., a finite set of measurements. Some of these measurements are generated by background

clutter Ω_k. Others are generated by the targets, with some targets possibly not having generated any. Mathematically speaking, Z is a random draw from an RFS Σ_k that can be decomposed as $\Sigma_k = \Upsilon(X_k) \cup \Omega_k$, where $\Upsilon(X_k)$ is the set of target-generated measurements.

Conventional Multitarget Detection and Tracking. This is based on a "divide and conquer" strategy with three basic steps: *time update*, *data association*, and *measurement update*. At time t_k we have n "tracks" $\tau_1, \ldots, \tau_n$ (hypothesized targets). In the time update, an extended Kalman filter (EKF) is used to time-predict the tracks τ_i to *predicted tracks* τ_i^+ at the time t_{k+1} of the next measurement set $Z_{k+1} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$.

Given Z_{k+1}, we can construct the following *data-association hypothesis H*: for each $i = 1, \ldots, n$, the predicted track τ_i^+ generated the detection $\mathbf{z}_{j_i}$, for some index j_i, or, alternatively, this track was not detected at all. If we remove from Z_{k+1} all of the $\mathbf{z}_{j_1}, \ldots, \mathbf{z}_{j_n}$, the remaining measurements are interpreted either as being clutter or as having been generated by new targets. Enumerating all possible association hypotheses (which is a combinatorily complex procedure), we end up with a "hypothesis table" $H_1, \ldots, H_\nu$.

Given H_i, let $\mathbf{z}_{j_i}$ be the measurement that is hypothesized to have been generated by predicted track τ_i^+. Then, the measurement-update step of an EKF is used to construct a measurement-updated track τ_{i,j_i} from τ_i^+ and $\mathbf{z}_{j_i}$. Attached to each H_i is a *hypothesis probability p_i* – the probability that the particular hypothesis H_i is the correct one. The hypothesis with largest p_i yields the multitarget estimate $\hat{X} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{\hat{n}}\}$.

RFS Multitarget Detection and Tracking. In the place of tracks and hypothesis tables, this uses *multitarget state sets* and *multitarget probability distributions*. In place of the conventional time update, data association, and measurement update, it uses a *recursive Bayes filter*. A random multitarget state set is an RFS $\Xi_{k|k}$ whose points are target states. A multitarget probability distribution is the probability distribution $f(X_k|Z_{1:k}) = f_{\Xi_{k|k}}(X)$ of the RFS $\Xi_{k|k}$,

where $Z_{1:k} : Z_1, \ldots, Z_k$ is the time sequence of measurement sets at time t_k.

RFS Time Update. The Bayes filter time-update step $f(X_k|Z_{1:k}) \rightarrow f(X_{k+1}|Z_{1:k})$ requires a *multitarget Markov transition function* $f(X_{k+1}|X_k)$. It is the probability that the multitarget system will have multitarget state set X_{k+1} at time t_{k+1}, if it had multitarget state set X_k at time t_k. It takes into account all pertinent characteristics of the targets: individual target motion, target appearance, target disappearance, environmental constraints, etc. It is explicitly constructed from an *RFS multitarget motion model* using a *multitarget integrodifferential calculus*.

RFS Measurement Update. The Bayes filter measurement-update step $f(X_{k+1}|Z_{1:k}) \rightarrow f(X_{k+1}|Z_{1:k+1})$ is just Bayes rule. It requires a *multitarget likelihood function* $f_{k+1}(Z|X)$ – the likelihood that a measurement set Z will be generated, if a system of targets with state set X is present. It takes into account all pertinent characteristics of the sensor(s): sensor noise, fields of view and obscurations, probabilities of detection, false alarms, and/or clutter. It is explicitly constructed from an *RFS measurement model* using multitarget calculus.

RFS State Estimation. Determination of the number n and states $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of the targets is accomplished using a *Bayes-optimal multitarget state estimator*. The idea is to determine the X_{k+1} that maximizes $f(X_{k+1}|Z_{1:k+1})$ in some sense.

Approximate Multitarget RFS Filters. The multitarget Bayes filter is, in general, computationally intractable. Central to the RFS approach is a toolbox of techniques – including the multitarget calculus – designed to produce *statistically principled* approximate multitarget filters. The two most well studied are the *probability hypothesis density* (PHD) *filter* and its generalization the *cardinalized PHD* (CPHD) *filter*. In such filters, $f(X_k|Z_{1:k})$ is replaced by the first-moment density $D(\mathbf{x}_k|Z_{1:k})$ of $\Xi_{k|k}$. These filters have been shown to be faster and

perform better than conventional approaches in some applications.

Random Sets and Human-Mediated Data

Random Sets and Human-Mediated Data

Natural-language statements and inference rules have already been mentioned as examples of human-mediated information. *Expert-systems theory* was introduced in part to address situations – such as this – that involve uncertainties other than randomness. Expert-system methodologies include *fuzzy set theory*, the *Dempster-Shafer* (D-S) *theory of uncertain evidence*, and *rule-based inference*. RS theory provides solid Bayesian foundations for them and allows human-mediated data to be processed using standard Bayesian estimation techniques. The purpose of this section is to briefly summarize this aspect of the RS approach.

The relationships between expert-systems theory and random set theory were first established by researchers such as Orlov (1978), Höhle (1982), Nguyen (1978), and Goodman and Nguyen (1985). At a relatively early stage, it was recognized that random set theory provided a potential means of unifying much of expert-systems theory (Goodman and Nguyen 1985; Kruse et al. 1991).

A conventional sensor measurement at time t_k is typically represented as $\mathbf{Z}_k = \eta(\mathbf{x}_k) + \mathbf{V}_k$ – equivalently formulated as a *likelihood function* $f(\mathbf{z}_k|\mathbf{x}_k)$. It is conventional to think of $\mathbf{z}_k$ as the actual "measurement" and of $f(\mathbf{z}_k|\mathbf{x}_k)$ as the full description of the uncertainty associated with it. In actuality, $\mathbf{z}_k$ is just a *mathematical model* $\mathbf{z}_{\zeta_k}$ of some *real-world measurement* ζ_k. Thus, the likelihood actually has the form $f(\zeta_k|\mathbf{x}_k) = f(\mathbf{z}_{\zeta_k}|\mathbf{x}_k)$.

This observation assumes crucial importance when one considers human-mediated data. Consider the simple natural-language statement

$$\zeta = \text{``The target is near the tower''}$$

where the *tower* is a landmark, located at a known position (x_0, y_0), and where the term "near" is assumed to have the following specific meaning: (x, y) is near (x_0, y_0) means that $(x, y) \in T_5$ where T_5 is a disk of radius $5\,$m, centered at (x_0, y_0). If $\mathbf{z} = (x, y)$ is the actual measurement of the target's position, then ζ is *equivalent to the formula* $\mathbf{z} \in T_5$. Since $\mathbf{z}$ is just one possible draw from $\mathbf{Z}_k$, we can say that ζ – or, equivalently, T_5 – is actually a constraint on the underlying measurement process: $\mathbf{Z}_k \in T_5$.

Because the word "near" is rather vague, we could just as well say that $\mathbf{z} \in T_5$ is the best choice, with confidence $w_5 = 0.7$; that $\mathbf{z} \in T_4$ is the next best choice, with confidence $w_4 = 0.2$; and that $\mathbf{z} \in T_6$ is the least best, with confidence $w_6 = 0.1$. Let Θ be the random subset of $\mathfrak{Z}$ defined by $\Pr(\Theta = T_i) = w_i$ for $i = 4, 5, 6$. In this case, ζ is equivalent to the *random constraint*

$$\mathbf{Z}_k \in \Theta.$$

The probability

$$\rho_k(\Theta | \mathbf{x}_k) = \Pr(\eta(\mathbf{x}_k) + \mathbf{V}_k \in \Theta)$$
$$= \Pr(\mathbf{Z}_k \in \Theta | \mathbf{X}_k = \mathbf{x}_k)$$

is called a *generalized likelihood function* (GLF). GLFs can be constructed for more complex natural-language statements, for inference rules, and more. Using their GLF representations, such "nontraditional measurements" can be processed using single- and multi-object recursive Bayes filters and their approximations. As a consequence, it can be shown that fuzzy logic, the D-S theory, and rule-based inference can be subsumed within a single Bayesian-probabilistic paradigm.

Summary and Future Directions

In the engineering world, the theory of random sets has been associated primarily with certain specialized image-processing applications, such as morphological filters and tomographic imaging. It has more recently found application in fields such as multitarget tracking and in expert-systems theory. All of these fields of application remain areas of active research.

Cross-References

▶ Estimation, Survey on
▶ Extended Kalman Filters
▶ Nonlinear Filters

Recommended Reading

Molchanov (2005) provides a definitive exposition of the general theory of random sets. Two excellent references for stochastic geometry are Stoyan et al. (1995) and Barndorff-Nielsen and van Lieshout (1999). The books by Kingman (1993) and Daley and Vere-Jones (1998) are good introductions to point process theory. The application of point process theory and stochastic geometry to image processing is addressed in, respectively, Snyder and Miller (1991) and Stoyan et al. (1995). The application of RFSs to multitarget estimation is addressed in the tutorials Mahler (2004, 2013) and the book Mahler (2007). Introductions to the application of random sets to expert systems can be found in Kruse et al. (1991) and Mahler (2007), Chaps. 3–6.

Bibliography

Barndorff-Nielsen O, van Lieshout M (1999) Stochastic geometry: likelihood and computation. Chapman/CRC, Boca Raton

Daley D, Vere-Jones D (1998) An introduction to the theory of point processes, 1st edn. Springer, New York

Goodman I, Nguyen H (1985) Uncertainty models for knowledge based systems. North-Holland, Amsterdam

Goutsias J, Mahler R, Nguyen H (eds) (1997) Random sets: theory and applications. Springer, New York

Höhle U (1982) A mathematical theory of uncertainty: fuzzy experiments and their realizations. In: Yager R (ed) Recent developments in fuzzy set and possibility theory. Pergamon, New York, pp 344–355

Kingman J (1993) Poisson processes. Oxford University Press, London

Kruse R, Schwencke E, Heinsohn J (1991) Uncertainty and vagueness in knowledge-based systems. Springer, New York

Mahler R (2004) 'Statistics 101' for multisensor, multitarget data fusion. IEEE Trans Aerosp Electron Sys Mag Part 2: Tutorials 19(1):53–64

Mahler R (2007) Statistical multisource-multitarget information fusion. Artech House, Norwood

Mahler R (2013) 'Statistics 102' for multisensor-multitarget tracking. IEEE J Spec Top Sign Proc 7(3):376–389

Matheron G (1975) Random sets and integral geometry. Wiley, New York

Michael E (1950) Topologies on spaces of subsets. Trans Am Math Soc 71:152–182

Molchanov I (2005) Theory of random sets. Springer, London

Nguyen H (1978) On random sets and belief functions. J Math Anal Appl 65:531–542

Orlov A (1978) Fuzzy and random sets. Prikladnoi Mnogomerni Statisticheskii Analys, Moscow

Snyder D, Miller M (1991) Random point processes in time and space, 2nd edn. Springer, New York

Stoyan D, Kendall W, Meche J (1995) Stochastic geometry and its applications, 2nd edn. Wiley, New York

Estimation, Survey on

Luigi Chisci[1] and Alfonso Farina[2]
[1]Dipartimento di Ingegneria dell'Informazione, Università di Firenze, Firenze, Italy
[2]Selex ES, Roma, Italy

Abstract

This entry discusses the history and describes the multitude of methods and applications of this important branch of stochastic process theory.

Keywords

Linear stochastic filtering; Markov step processes; Maximum likelihood estimation; Riccati equation; Stratonovich-Kushner equation

Estimation is the process of inferring the value of an unknown given quantity of interest from noisy, direct or indirect, observations of such a quantity. Due to its great practical relevance, estimation has a long history and an enormous variety of applications in all fields of engineering and science. A certainly incomplete list of possible application domains of estimation includes the following: statistics (Bard 1974; Ghosh et al. 1997; Koch 1999; Lehmann and Casella 1998; Tsybakov 2009; Wertz 1978), telecommunication systems (Sage and Melsa 1971; Schonhoff and Giordano 2006; Snyder 1968; Van Trees 1971), signal and image processing (Barkat 2005; Biemond et al. 1983; Elliott et al. 2008; Itakura 1971; Kay 1993; Kim and Woods 1998; Levy 2008; Najim 2008; Poor 1994; Tuncer and Friedlander 2009; Wakita 1973; Woods and Radewan 1977), aerospace engineering (McGee and Schmidt 1985), tracking (Bar-Shalom and Fortmann 1988; Bar-Shalom et al. 2001, 2013; Blackman and Popoli 1999; Farina and Studer 1985, 1986), navigation (Dissanayake et al. 2001; Durrant-Whyte and Bailey 2006a,b; Farrell and Barth 1999; Grewal et al. 2001; Mullane et al. 2011; Schmidt 1966; Smith et al. 1986; Thrun et al. 2006), control systems (Anderson and Moore 1979; Athans 1971; Goodwin et al. 2005; Joseph and Tou 1961; Kalman 1960a; Maybeck 1979, 1982; Söderström 1994; Stengel 1994), econometrics (Aoki 1987; Pindyck and Roberts 1974; Zellner 1971), geophysics (e.g., seismic deconvolution) (Bayless and Brigham 1970; Flinn et al. 1967; Mendel 1977, 1983, 1990), oceanography (Evensen 1994a; Ghil and Malanotte-Rizzoli 1991), weather forecasting (Evensen 1994b, 2007; McGarty 1971), environmental engineering (Dochain and Vanrolleghem 2001; Heemink and Segers 2002; Nachazel 1993), demographic systems (Leibungudt et al. 1983), automotive systems (Barbarisi et al. 2006; Stephant et al. 2004), failure detection (Chen and Patton 1999; Mangoubi 1998; Willsky 1976), power systems (Abur and Gómez Espósito 2004; Debs and Larson 1970; Miller and Lewis 1971; Monticelli 1999; Toyoda et al. 1970), nuclear engineering (Robinson 1963; Roman et al. 1971; Sage and Masters 1967; Venerus and Bullock 1970), biomedical engineering (Bekey 1973; Snyder 1970; Stark 1968), pattern recognition (Andrews 1972; Ho and Agrawala 1968; Lainiotis 1972), social networks (Snijders et al. 2012), etc.

Chapter Organization

The rest of the chapter is organized as follows. Section "Historical Overview on Estimation" will provide a historical overview on estimation. The next section will discuss applications of estimation. Connections between estimation and information theories will be explored in the subsequent section. Finally, the section "Conclusions and Future Trends" will conclude the chapter by discussing future trends in estimation. An extensive list of references is also provided.

Historical Overview on Estimation

A possibly incomplete, list of the major achievements on estimation theory and applications is reported in Table 1. The entries of the table, sorted in chronological order, provide for each contribution the name of the inventor (or inventors), the date, and a short description with main bibliographical references.

Probably the first important application of estimation dates back to the beginning of the nineteenth century whenever least-squares estimation (LSE), invented by Gauss in 1795 (Gauss 1995; Legendre 1810), was successfully exploited in astronomy for predicting planet orbits (Gauss 1806). Least-squares estimation follows a deterministic approach by minimizing the sum of squares of residuals defined as differences between observed data and model-predicted estimates. A subsequently introduced statistical approach is maximum likelihood estimation (MLE), popularized by R. A. Fisher between 1912 and 1922 (Fisher 1912, 1922, 1925). MLE consists of finding the estimate of the unknown quantity of interest as the value that maximizes the so-called likelihood function, defined as the conditional probability density function of the observed data given the quantity to be estimated. In intuitive terms, MLE maximizes the agreement of the estimate with the observed data. Whenever the observation noise is assumed Gaussian (Kim and Shevlyakov 2008; Park et al. 2013), MLE coincides with LSE.

While estimation problems had been addressed for several centuries, it was not until the 1940s that a systematic theory of estimation started to be established, mainly relying on the foundations of the modern theory of probability (Kolmogorov 1933). Actually, the roots of probability theory can be traced back to the calculus of combinatorics (the Stomachion puzzle invented by Archimedes (Netz and Noel 2011)) in the third century B.C. and to the gambling theory (work of Cardano, Pascal, de Fermat, Huygens) in the sixteenth–seventeenth centuries.

Differently from the previous work devoted to the estimation of constant parameters, in the period 1940–1960 the attention was mainly shifted toward the estimation of signals. In particular, Wiener in 1940 (Wiener 1949) and Kolmogorov in 1941 (Kolmogorov 1941) formulated and solved the problem of linear minimum mean-square error (MMSE) estimation of continuous-time and, respectively, discrete-time stationary random signals. In the late 1940s and in the 1950s, Wiener-Kolmogorov's theory was extended and generalized in many directions exploiting both time-domain and frequency-domain approaches. At the beginning of the 1960s Rudolf E. Kálmán made pioneering contributions to estimation by providing the mathematical foundations of the modern theory based on state-variable representations. In particular, Kálmán solved the linear MMSE filtering and prediction problems both in discrete-time (Kalman 1960b) and in continuous-time (Kalman and Bucy 1961); the resulting optimal estimator was named after him, Kalman filter (KF). As a further contribution, Kalman also singled out the key technical conditions, i.e., observability and controllability, for which the resulting optimal estimator turns out to be stable. Kalman's work went well beyond earlier contributions of A. Kolmogorov, N. Wiener, and their followers ("frequency-domain" approach) by means of a general state-space approach. From the theoretical viewpoint, the KF is an optimal estimator, in a wide sense, of the state of a linear dynamical system from noisy measurements; specifically it is the optimal MMSE estimator in

Estimation, Survey on, Table 1 Major developments on estimation

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
G. Cardano, B. Pascal, P. de Fermat, C. Huygens	Sixteenth–seventeenth centuries	Roots of the theory of probability (Devlin 2008)
J. F. Riccati	1722–1723	Differential Riccati equation (Riccati 1722, 1723), subsequently exploited in the theory of linear stochastic filtering
T. Bayes	1763	Bayes' formula on conditional probability (Bayes 1763; McGrayne 2011)
C. F. Gauss, A. M. Legendre	1795–1810	Least-squares estimation and its applications to the prediction of planet orbits (Gauss 1806, 1995; Legendre 1810)
P. S. Laplace	1814	Theory of probability (Laplace 1814)
R. A. Fisher	1912–1922	Maximum likelihood estimation (Fisher 1912, 1922, 1925)
A. N. Kolmogorov	1933	Modern theory of probability (Kolmogorov 1933)
N. Wiener	1940	Minimum mean-square error estimation of continuous-time stationary random signals (Wiener 1949)
A. N. Kolmogorov	1941	Minimum mean-square error estimation of discrete-time stationary random signals (Kolmogorov 1941)
H. Cramér, C. R. Rao	1945	Theoretical lower bound on the covariance of estimators (Cramér 1946; Rao 1945)
S. Ulam, J. von Neumann, N. Metropolis, E. Fermi	1946–1949	Monte Carlo method (Los Alamos Scientific Laboratory 1966; Metropolis and Ulam 1949; Ulam 1952; Ulam et al. 1947)
J. Sklansky, T. R. Benedict, G. W. Bordner, H. R. Simpson, S. R. Neal	1957–1967	$\alpha - \beta$ and $\alpha - \beta - \gamma$ filters (Benedict and Bordner 1962; Neal 1967; Painter et al. 1990; Simpson 1963; Sklansky 1957)
R. L. Stratonovich, H. J. Kushner	1959–1964	Bayesian approach to stochastic nonlinear filtering of continuous-time systems, i.e., Stratonovich-Kushner equation for the evolution of the state conditional probability density (Jazwinski 1970; Kushner 1962, 1967; Stratonovich 1959, 1960)
R. E. Kalman	1960	Linear filtering and prediction for discrete-time systems (Kalman 1960b)
R. E. Kalman	1961	Observability of linear dynamical systems (Kalman 1960a)
R. E. Kalman, R. S. Bucy	1961	Linear filtering and prediction for continuous-time systems (Kalman and Bucy 1961)
A. E. Bryson, M. Frazier, H. E. Rauch, F. Tung, C. T. Striebel, D. Q. Mayne, J. S. Meditch, D. C. Fraser, L. E. Zachrisson, B. D. O. Anderson, etc.	Since 1963	Smoothing of linear and nonlinear systems (Anderson and Chirarattananon 1972; Bryson and Frazier 1963; Mayne 1966; Meditch 1967; Rauch 1963; Rauch et al. 1965; Zachrisson 1969)
D. G. Luenberger	1964	State observer for a linear system (Luenberger 1964)
Y. C. Ho, R. C. K. Lee	1964	Bayesian approach to recursive nonlinear estimation for discrete-time systems (Ho and Lee 1964)
W. M. Wonham	1965	Optimal filtering for Markov step processes (Wonham 1965)
A. H. Jazwinski	1966	Bayesian approach to stochastic nonlinear filtering for continuous-time stochastic systems with discrete-time observations (Jazwinski 1966)

Estimation, Survey on, Table 1 (continued)

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
S. F. Schmidt	1966	Extended Kalman filter and its application for the manned lunar missions (Schmidt 1966)
P. L. Falb, A. V. Balakrishnan, J. L. Lions, S. G. Tzafestas, J. M. Nightingale, H. J. Kushner, J. S. Meditch, etc.	Since 1967	State estimation for infinite-dimensional (e.g., distributed parameter, partial differential equation (PDE), delay) systems (Balakrishnan and Lions 1967; Falb 1967; Kushner 1970; Kwakernaak 1967; Meditch 1971; Tzafestas and Nightingale 1968)
T. Kailath	1968	Principle of orthogonality and innovation approach to estimation (Frost and Kailath 1971; Kailath 1968, 1970; Kailath and Frost 1968; Kailath et al. 2000)
A. H. Jazwinski, B. Rawlings, etc.	Since 1968	Limited memory (receding-horizon, moving-horizon) state estimation with constraints (Alessandri et al. 2005, 2008; Jazwinski 1968; Rao et al. 2001, 2003)
F. C. Schweppe, D. P. Bertsekas, I. B. Rhodes, M. Milanese, etc.	Since 1968	Set-membership recursive state estimation with systems with unknown but bounded noises (Alamo et al. 2005; Bertsekas and Rhodes 1971; Chisci et al. 1996; Combettes 1993; Milanese and Belforte 1982; Milanese and Vicino 1993; Schweppe 1968; Vicino and Zappa 1996)
J. E. Potter, G. Golub, S. F. Schmidt, P. G. Kaminski, A. E. Bryson, A. Andrews, G. J. Bierman, M. Morf, T. Kailath, etc.	1968–1975	Square-root filtering (Andrews 1968; Bierman 1974, 1977; Golub 1965; Kaminski and Bryson 1972; Morf and Kailath 1975; Potter and Stern 1963; Schmidt 1970)
C. W. Helstrom	1969	Quantum estimation (Helstrom 1969, 1976)
D. L. Alspach, H. W. Sorenson	1970–1972	Gaussian-sum filters for nonlinear and/or non-Gaussian systems (Alspach and Sorenson 1972; Sorenson and Alspach 1970, 1971)
T. Kailath, M. Morf, G. S. Sidhu	1973–1974	Fast Chandrasekhar-type algorithms for recursive state estimation of stationary linear systems (Kailath 1973; Morf et al. 1974)
A. Segall	1976	Recursive estimation from point processes (Segall 1976)
J. W. Woods and C. Radewan	1977	Kalman filter in two dimensions (Woods and Radewan 1977) for image processing
J. H. Taylor	1979	Cramér-Rao lower bound (CRLB) for recursive state estimation with no process noise (Taylor 1979)
D. Reid	1979	Multiple Hypothesis Tracking (MHT) filter for multi-target tracking (Reid 1979)
L. Servi, Y. Ho	1981	Optimal filtering for linear systems with uniformly distributed measurement noise (Servi and Ho 1981)
V. E. Benes	1981	Exact finite-dimensional optimal MMSE filter for a class of nonlinear systems (Benes 1981)
H. V. Poor, D. Looze, J. Darragh, S. Verdú, M. J. Grimble, etc.	1981–1988	Robust (e.g., H_∞) filtering (Darragh and Looze 1984; Grimble 1988; Hassibi et al. 1999; Poor and Looze 1981; Simon 2006; Verdú and Poor 1984)
V. J. Aidala, S. E. Hammel	1983	Bearings-only tracking (Aidala and Hammel 1983; Farina 1999)
F. E. Daum	1986	Extension of the Benes filter to a more general class of nonlinear systems (Daum 1986)

(continued)

Estimation, Survey on, Table 1 (continued)

Archimedes	Third century B.C.	Combinatorics (Netz and Noel 2011) as the basis of probability
L. Dai and others	Since 1987	State estimation for linear descriptor (singular, implicit) stochastic systems (Chisci and Zappa 1992; Dai 1987, 1989; Nikoukhah et al. 1992)
N. J. Gordon, D. J. Salmond, A. M. F. Smith	1993	Particle (sequential Monte Carlo) filter (Doucet et al. 2001; Gordon et al. 1993; Ristic et al. 2004)
K. C. Chou, A. S. Willsky, A. Benveniste	1994	Multiscale Kalman filter (Chou et al. 1994)
G. Evensen	1994	Ensemble Kalman filter for data assimilation in meteorology and oceanography (Evensen 1994b, 2007)
R. P. S. Mahler	1994	Random set filtering (Mahler 1994, 2007a; Ristic et al. 2013)
S. J. Julier, J. K. Uhlmann, H. Durrant-Whyte	1995	Unscented Kalman filter (Julier and Uhlmann 2004; Julier et al. 1995)
A. Germani et al.	Since 1996	Polynomial extended Kalman filter for nonlinear and/or non-Gaussian systems (Carravetta et al. 1996; Germani et al. 2005)
P. Tichavsky, C. H. Muravchik, A. Nehorai	1998	Posterior Cramér-Rao lower bound (PCRLB) for recursive state estimation (Tichavsky et al. 1998; van Trees and Bell 2007)
R. Mahler	2003, 2007	Probability hypothesis density (PHD) and cardinalized PHD (CPHD) filters (Mahler 2003, 2007b; Ristic 2013; Vo and Ma 1996; Vo et al. 2007)
A.G. Ramm	2005	Estimation of random fields (Ramm 2005)
M. Hernandez, A. Farina, B. Ristic	2006	PCRLB for tracking in the case of detection probability less than one and false alarm probability greater than zero (Hernandez et al. 2006)
Olfati-Saber and others	Since 2007	Consensus filters (Olfati-Saber et al. 2007; Calafiore and Abrate 2009; Xiao et al. 2005; Alriksson and Rantzer 2006; Olfati-Saber 2007; Kamgarpour and Tomlin 2007; Stankovic et al. 2009; Battistelli et al. 2011, 2012, 2013; Battistelli and Chisci 2014) for networked estimation

the Gaussian case (e.g., for normally distributed noises and initial state) and the best linear unbiased estimator irrespective of the noise and initial state distributions. From the practical viewpoint, the KF enjoys the desirable properties of being linear and acting recursively, step-by-step, on a noise-contaminated data stream. This allows for cheap real-time implementation on digital computers. Further, the universality of "state-variable representations" allows almost any estimation problem to be included in the KF framework. For these reasons, the KF is, and continues to be, an extremely effective and easy-to-implement tool for a great variety of practical tasks, e.g., to detect signals in noise or to estimate unmeasurable quantities from accessible observables. Due to the generality of the state estimation problem, which actually encompasses parameter and signal estimation as special cases, the literature on estimation since 1960 till today has been mostly concentrated on extensions and generalizations of Kalman's work in several directions. Considerable efforts, motivated by the ubiquitous presence of nonlinearities in practical estimation problems, have been devoted to nonlinear and/or non-Gaussian filtering, starting from the seminal papers of Stratonovich (1959, 1960) and Kushner (1962, 1967) for continuous-time systems, Ho and Lee (1964) for discrete-time systems, and Jazwinski (1966) for continuous-time systems with discrete-time observations. In these

papers, state estimation is cast in a probabilistic (Bayesian) framework as the problem of evolving in time the state conditional probability density given observations (Jazwinski 1970). Work on nonlinear filtering has produced over the years several nonlinear state estimation algorithms, e.g., the extended Kalman filter (EKF) (Schmidt 1966), the unscented Kalman filter (UKF) (Julier and Uhlmann 2004; Julier et al. 1995), the Gaussian-sum filter (Alspach and Sorenson 1972; Sorenson and Alspach 1970, 1971), the sequential Monte Carlo (also called particle) filter (SMCF) (Doucet et al. 2001; Gordon et al. 1993; Ristic et al. 2004), and the ensemble Kalman filter (EnKF) (Evensen 1994a,b, 2007) which have been, and are still now, successfully employed in various application domains. In particular, the SMCF and EnKF are stochastic simulation algorithms taking inspiration from the work in the 1940s on the Monte Carlo method (Metropolis and Ulam 1949) which has recently got renewed interest thanks to the tremendous advances in computing technology. A thorough review on nonlinear filtering can be found, e.g., in Daum (2005) and Crisan and Rozovskii (2011).

Other interesting areas of investigation have concerned smoothing (Bryson and Frazier 1963), robust filtering for systems subject to modeling uncertainties (Poor and Looze 1981), and state estimation for infinite-dimensional (i.e., distributed parameter and/or delay) systems (Balakrishnan and Lions 1967). Further, a lot of attention has been devoted to the implementation of the KF, specifically square-root filtering (Potter and Stern 1963) for improved numerical robustness and fast KF algorithms (Kailath 1973; Morf et al. 1974) for enhancing computational efficiency. Worth of mention is the work over the years on theoretical bounds on the estimation performance originated from the seminal papers of Rao (1945) and Cramér (1946) on the lower bound of the MSE for parameter estimation and subsequently extended in Tichavsky et al. (1998) to nonlinear filtering and in Hernandez et al. (2006) to more realistic estimation problems with possible missed and/or false measurements. An extensive review of this work on Bayesian bounds for estimation, nonlinear filtering, and tracking

can be found in van Trees and Bell (2007). A brief review of the earlier (until 1974) state of art in estimation can be found in Lainiotis (1974).

Applications

Astronomy

The problem of making estimates and predictions on the basis of noisy observations originally attracted the attention many centuries ago in the field of astronomy. In particular, the first attempt to provide an optimal estimate, i.e., such that a certain measure of the estimation error be minimized, was due to Galileo Galilei that, in his *Dialogue on the Two World Chief Systems* (1632) (Galilei 1632), suggested, as a possible criterion for estimating the position of Tycho Brahe's supernova, the estimate that required the "minimum amendments and smallest corrections" to the data. Later, C. F. Gauss mathematically specified this criterion by introducing in 1795 the least-squares method (Gauss 1806, 1995; Legendre 1810) which was successfully applied in 1801 to predict the location of the asteroid Ceres. This asteroid, originally discovered by the Italian astronomer Giuseppe Piazzi on January 1, 1801, and then lost in the glare of the sun, was in fact recovered 1 year later by the Hungarian astronomer F. X. von Zach exploiting the least-squares predictions of Ceres' position provided by Gauss.

Statistics

Starting from the work of Fisher in the 1920s (Fisher 1912, 1922, 1925), maximum likelihood estimation has been extensively employed in statistics for estimating the parameters of statistical models (Bard 1974; Ghosh et al. 1997; Koch 1999; Lehmann and Casella 1998; Tsybakov 2009; Wertz 1978).

Telecommunications and Signal/Image Processing

Wiener-Kolmogorov's theory on signal estimation, developed in the period 1940–1960 and originally conceived by Wiener during the Second World War for predicting aircraft

trajectories in order to direct the antiaircraft fire, subsequently originated many applications in telecommunications and signal/image processing (Barkat 2005; Biemond et al. 1983; Elliott et al. 2008; Itakura 1971; Kay 1993; Kim and Woods 1998; Levy 2008; Najim 2008; Poor 1994; Tuncer and Friedlander 2009; Van Trees 1971; Wakita 1973; Woods and Radewan 1977). For instance, Wiener filters have been successfully applied to linear prediction, acoustic echo cancellation, signal restoration, and image/video de-noising. But it was the discovery of the Kalman filter in 1960 that revolutionized estimation by providing an effective and powerful tool for the solution of any, static or dynamic, stationary or adaptive, linear estimation problem. A recently conducted, and probably non-exhaustive, search has detected the presence of over 16,000 patents related to the "Kalman filter," spreading over all areas of engineering and over a period of more than 50 years. What is astonishing is that even nowadays, more than 50 years after its discovery, one can see the continuous appearance of lots of new patents and scientific papers presenting novel applications and/or novel extensions in many directions (e.g., to nonlinear filtering) of the KF. Since 1992 the number of patents registered every year and related to the KF follows an exponential law.

Space Navigation and Aerospace Applications

The first important application of the Kalman filter was in the NASA (*National Aeronautic and Space Administration*) space program. As reported in a NASA technical report (McGee and Schmidt 1985), Kalman presented his new ideas while visiting Stanley F. Schmidt at the NASA Ames Research Center in 1960, and this meeting stimulated the use of the KF during the Apollo program (in particular, in the guidance system of Saturn V during Apollo 11 flight to the Moon), and, furthermore, in the NASA Space Shuttle and in Navy submarines and unmanned aerospace vehicles and weapons, such as cruise missiles. Further, to cope with the nonlinearity of the space navigation problem and the small word length of the onboard computer, the extended Kalman

filter for nonlinear systems and square-root filter implementations for enhanced numerical robustness have been developed as part of the NASA's Apollo program. The aerospace field was only the first of a long and continuously expanding list of application domains where the Kalman filter and its nonlinear generalizations have found widespread and beneficial use.

Control Systems and System Identification

The work on Kalman filtering (Kalman 1960b; Kalman and Bucy 1961) had also a significant impact on control system design and implementation. In Kalman (1960a) duality between estimation and control was pointed out, in that for a certain class of control and estimation problems one can solve the control (estimation) problem for a given dynamical system by resorting to a corresponding estimation (control) problem for a suitably defined dual system. In particular, the Kalman filter has been shown to be dual of the linear-quadratic (LQ) regulator, and the two dual techniques constitute the linear-quadratic-Gaussian (LQG) (Joseph and Tou 1961) regulator. The latter consists of an LQ regulator feeding back in a linear way the state estimate provided by a Kalman filter, which can be independently designed in view of the separation principle. The KF as well as LSE and MLE techniques are also widely used in system identification (Ljung 1999; Söderström and Stoica 1989) for both parameter estimation and output prediction purposes.

Tracking

One of the major application areas for estimation is tracking (Bar-Shalom and Fortmann 1988; Bar-Shalom et al. 2001, 2013; Blackman and Popoli 1999; Farina and Studer 1985, 1986), i.e., the task of following the motion of moving objects (e.g., aircrafts, ships, ground vehicles, persons, animals) given noisy measurements of kinematic variables from remote sensors (e.g., radar, sonar, video cameras, wireless sensors, etc.). The development of the Wiener filter in the 1940s was actually motivated by radar tracking of aircraft for automatic control of antiaircraft guns. Such filters began to be used in the 1950s whenever

computers were integrated with radar systems, and then in the 1960s more advanced and better performing Kalman filters came into use. Still today it can be said that the Kalman filter and its nonlinear generalizations (e.g., EKF (Schmidt 1966), UKF (Julier and Uhlmann 2004), and particle filter (Gordon et al. 1993)) represent the workhorses of tracking and sensor fusion. Tracking, however, is usually much more complicated than a simple state estimation problem due to the presence of false measurements (clutter) and multiple objects in the surveillance region of interest, as well as for the uncertainty about the origin of measurements. This requires to use, besides filtering algorithms, smart techniques for object detection as well as for association between detected objects and measurements. The problem of joint target tracking and classification has also been formulated as a hybrid state estimation problem and addressed in a number of papers (see, e.g., Smeth and Ristic (2004) and the references therein).

Econometrics

State and parameter estimation have been widely used in econometrics (Aoki 1987) for analyzing and/or predicting financial time series (e.g., stock prices, interest rates, unemployment rates, volatility etc.).

Geophysics

Wiener and Kalman filtering techniques are employed in reflection seismology for estimating the unknown earth reflectivity function given noisy measurements of the seismic wavelet's echoes recorded by a geophone. This estimation problem, known as seismic deconvolution (Mendel 1977, 1983, 1990), has been successfully exploited, e.g., for oil exploration.

Data Assimilation for Weather Forecasting and Oceanography

Another interesting application of estimation theory is data assimilation (Ghil and Malanotte-Rizzoli 1991) which consists of incorporating noisy observations into a computer simulation model of a real system. Data assimilation has widespread use especially in weather forecasting and oceanography. A large-scale state-space model is typically obtained from the physical system model, expressed in terms of partial differential equations (PDEs), by means of a suitable spatial discretization technique so that data assimilation is cast into a state estimation problem. To deal with the huge dimensionality of the resulting state vector, appropriate filtering techniques with reduced computational load have been suitably developed (Evensen 2007).

Global Navigation Satellite Systems

Global Navigation Satellite Systems (GNSSs), such as GPS put into service in 1993 by the US Department of Defense, provide nowadays a commercially diffused technology exploited by millions of users all over the world for navigation purposes, wherein the Kalman filter plays a key role (Bar-Shalom et al. 2001). In fact, the Kalman filter not only is employed in the core of the GNSS to estimate the trajectories of all the satellites, the drifts and rates of all system clocks, and hundreds of parameters related to atmospheric propagation delay, but also any GNSS receiver uses a nonlinear Kalman filter, e.g., EKF, in order to estimate its own position and velocity along with the bias and drift of its own clock with respect to the GNSS time.

Robotic Navigation (SLAM)

Recursive state estimation is commonly employed in mobile robotics (Thrun et al. 2006) in order to on-line estimate the robot pose, location and velocity, and, sometimes, also the location and features of the surrounding objects in the environment exploiting measurements provided by onboard sensors; the overall joint estimation problem is referred to as SLAM (simultaneous localization and mapping) (Dissanayake et al. 2001; Durrant-Whyte and Bailey 2006a,b; Mullane et al. 2011; Smith et al. 1986; Thrun et al. 2006).

Automotive Systems

Several automotive applications of the Kalman filter, or of its nonlinear variants, are reported in the literature for the estimation of various

quantities of interest that cannot be directly measured, e.g., roll angle, sideslip angle, road-tire forces, heading direction, vehicle mass, state of charge of the battery (Barbarisi et al. 2006), etc. In general, one of the major applications of state estimation is the development of virtual sensors, i.e., estimation algorithms for physical variables of interest, that cannot be directly measured for technical and/or economic reasons (Stephant et al. 2004).

Miscellaneous Applications

Other areas where estimation has found numerous applications include electric power systems (Abur and Gómez Espósito 2004; Debs and Larson 1970; Miller and Lewis 1971; Monticelli 1999; Toyoda et al. 1970), nuclear reactors (Robinson 1963; Roman et al. 1971; Sage and Masters 1967; Venerus and Bullock 1970), biomedical engineering (Bekey 1973; Snyder 1970; Stark 1968), pattern recognition (Andrews 1972; Ho and Agrawala 1968; Lainiotis 1972), and many others.

Connection Between Information and Estimation Theories

In this section, the link between two fundamental quantities in information theory and estimation theory, i.e., the mutual information (MI) and respectively the minimum mean-square error (MMSE), is investigated. In particular, a strikingly simple but very general relationship can be established between the MI of the input and the output of an additive Gaussian channel and the MMSE in estimating the input given the output, regardless of the input distribution (Guo et al. 2005). Although this functional relation holds for general settings of the Gaussian channel (e.g., both discrete-time and continuous-time, possibly vector, channels), in order to avoid the heavy mathematical preliminaries needed to treat rigorously the general problem, two simple scalar cases, a static and a (continuous-time) dynamic one, will be discussed just to highlight the main concept.

Static Scalar Case

Consider two scalar real-valued random variables, x and y, related by

$$y = \sqrt{\sigma}x + v \qquad (1)$$

where v, the measurement noise, is a standard Gaussian random variable independent of x and σ can be regarded as the gain in the output signal-to-noise ratio (SNR) due to the channel. By considering the MI between x and y as a function of σ, i.e., $I(\sigma) = I\left(x, \sqrt{\sigma}x + v\right)$, it can be shown that the following relation holds (Guo et al. 2005):

$$\frac{d}{d\sigma}I(\sigma) = \frac{1}{2}\,E\left[(x - \hat{x}(\sigma))^2\right] \qquad (2)$$

where $\hat{x}(\sigma) = E\left[x|\sqrt{\sigma}x + v\right]$ is the minimum mean-square error estimate of x given y. Figure 1 displays the behavior of both MI, in natural logarithmic units of information (nats), and MMSE versus SNR.

As mentioned in Guo et al. (2005), the above information-estimation relationship (2) has found a number of applications, e.g., in nonlinear filtering, in multiuser detection, in power allocation over parallel Gaussian channels, in the proof of Shannon's entropy power inequality and its generalizations, as well as in the treatment of the capacity region of several multiuser channels.

Linear Dynamic Continuous-Time Case

While in the static case the MI is assumed to be a function of the SNR, in the dynamic case it is of great interest to investigate the relationship between the MI and the MMSE as a function of time.

Consider the following first-order (scalar) linear Gaussian continuous-time stochastic dynamical system:

$$\begin{aligned} dx_t &= ax_t dt + dw_t \\ dy_t &= \sqrt{\sigma}x_t\, dt + dv_t \end{aligned} \qquad (3)$$

where a is a real-valued constant while w_t and v_t are independent standard Brownian motion processes that represent the process and,

E
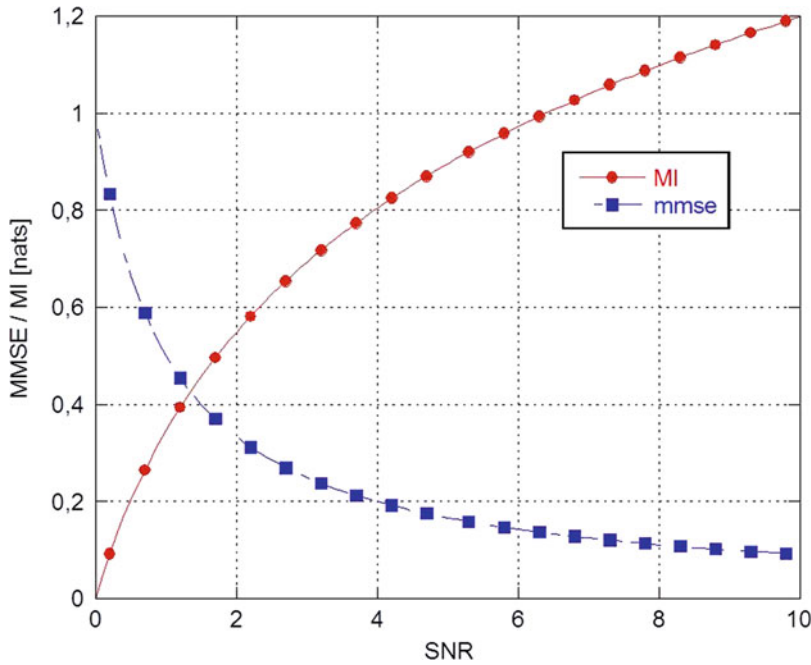
Estimation, Survey on, Fig. 1 MI and MMSE versus SNR

respectively, measurement noises. Defining by $x_0^t \triangleq \{x_s, 0 \leq s \leq t\}$ the collection of all states up to time t and analogously $y_0^t \triangleq \{y_s, 0 \leq s \leq t\}$ for the channel outputs (i.e., measurements) and considering the MI between x_0^t and y_0^t as a function of time t, i.e., $I(t) = I\left(x_0^t, y_0^t\right)$, it can be shown that (Duncan 1970; Mayer-Wolf and Zakai 1983)

$$\frac{d}{dt} I(t) = \frac{\sigma}{2} E\left[(x_t - \hat{x}_t)^2\right] \qquad (4)$$

where $\hat{x}_t = E\left[x_t | y_0^t\right]$ is the minimum mean-square error estimate of the state x_t given all the channel outputs up to time t, i.e., y_0^t. Figure 2 depicts the time behavior of both MI and MMSE for several values of σ and $a = 1$.

Conclusions and Future Trends

Despite the long history of estimation and the huge amount of work on several theoretical and practical aspects of estimation, there is still a lot of research investigation to be done in several directions. Among the many new future trends, networked estimation and quantum estimation (briefly overviewed in the subsequent parts of this section) certainly deserve special attention due to the growing interest on wireless sensor networks and, respectively, quantum computing.

Networked Information Fusion and Estimation

Information or data fusion is about combining, or fusing, information or data from multiple sources to provide knowledge that is not evident from a single source (Bar-Shalom et al. 2013; Farina and Studer 1986). In 1986, an effort to standardize the terminology related to data fusion began and the JDL (Joint Directors of Laboratories) data fusion working group was established. The result of that effort was the conception of a process model for data fusion and a data fusion lexicon (Blasch et al. 2012; Hall and Llinas 1997). Information and data fusion are mainly supported by sensor networks which present the following advantages over a single sensor:
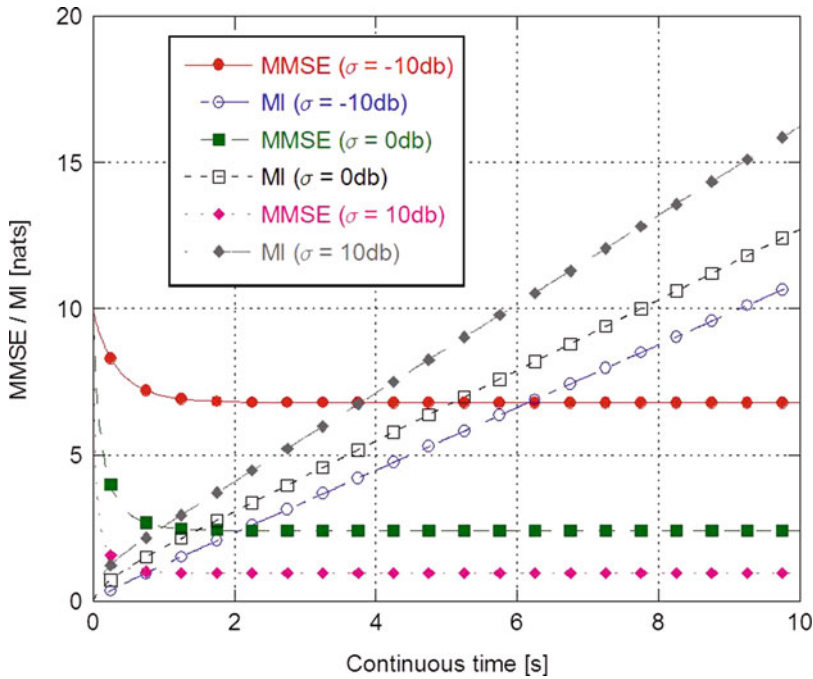
Estimation, Survey on, Fig. 2 MI and MMSE for different values of σ (-10 dB, 0 dB, $+10$ dB) and $a = 1$

- Can be deployed over wide regions
- Provide diverse characteristics/viewing angles of the observed phenomenon
- Are more robust to failures
- Gather more data that, once fused, provide a more complete picture of the observed phenomenon
- Allow better geographical coverage, i.e., wider area and less terrain obstructions.

Sensor network architectures can be centralized, hierarchical (with or without feedback), and distributed (peer-to-peer). Today's trend for many monitoring and decision-making tasks is to exploit large-scale networks of low-cost and low-energy consumption devices with sensing, communication, and processing capabilities. For scalability issues, such networks should operate in a fully distributed (peer-to-peer) fashion, i.e., with no centralized coordination, so as to achieve in each node a global estimation/decision objective through localized processing only.

The attainment of this goal actually requires several issues to be addressed like:

- Spatial and temporal sensor alignment
- Scalable fusion
- Robustness with respect to data incest (or double counting), i.e., repeated use of the same information
- Handling data latency (e.g., out-of-sequence measurements/estimates)
- Communication bandwidth limitations

In particular, to counteract data incest the so-called *covariance intersection* (Julier and Uhlmann 1997) robust fusion approach has been proposed to guarantee, at the price of some conservatism, consistency of the fused estimate when combining estimates from different nodes with unknown correlations. For scalable fusion, a consensus approach (Olfati-Saber et al. 2007) can be undertaken. This allows to carry out a global (i.e., over the whole network) processing task by iterating local processing steps among neighboring nodes.

Several consensus algorithms have been proposed for distributed parameter (Calafiore and Abrate 2009) or state (Alriksson and

Rantzer 2006; Kamgarpour and Tomlin 2007; Olfati-Saber 2007; Stankovic et al. 2009; Xiao et al. 2005) estimation. Recently, Battistelli and Chisci (2014) introduced a generalized consensus on probability densities which opens up the possibility to perform in a fully distributed and scalable way any Bayesian estimation task over a sensor network. As by-products, this approach allowed to derive consensus Kalman filters with guaranteed stability under minimal requirements of system observability and network connectivity (Battistelli et al. 2011, 2012; Battistelli and Chisci 2014), consensus nonlinear filters (Battistelli et al. 2012), and a consensus CPHD filter for distributed multitarget tracking (Battistelli et al. 2013). Despite these interesting preliminary results, networked estimation is still a very active research area with many open problems related to energy efficiency, estimation performance optimality, robustness with respect to delays and/or data losses, etc.

Quantum Estimation

Quantum estimation theory consists of a generalization of the classical estimation theory in terms of quantum mechanics. As a matter of fact, the statistical theory can be seen as a particular case of the more general quantum theory (Helstrom 1969, 1976). Quantum mechanics presents practical applications in several fields of technology (Personick 1971) such as, the use of quantum number generators in place of the classical random number generators. Moreover, manipulating the energy states of the cesium atoms, it is possible to suppress the quantum noise levels and consequently improve the accuracy of atomic clocks. Quantum mechanics can also be exploited to solve optimization problems, giving sometimes optimization algorithms that are faster than conventional ones. For instance, McGeoch and Wang (2013) provided an experimental study of algorithms based on quantum annealing. Interestingly, the results of McGeoch and Wang (2013) have shown that this approach allows to obtain better solutions with respect to those found with conventional software solvers. In quantum mechanics, also the Kalman filter has found its proper form, as the quantum Kalman filter.

In Iida et al. (2010) the quantum Kalman filter is applied to an optical cavity composed of mirrors and crystals inside, which interacts with a probe laser. In particular, a form of a quantum stochastic differential equation can be written for such a system so as to design the algorithm that updates the estimates of the system variables on the basis of the measurement outcome of the system.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Bounds on Estimation
▶ Consensus of Complex Multi-agent Systems
▶ Data Association
▶ Estimation and Control over Networks
▶ Estimation for Random Sets
▶ Extended Kalman Filters
▶ Kalman Filters
▶ Moving Horizon Estimation
▶ Networked Control Systems: Estimation and Control over Lossy Networks
▶ Nonlinear Filters
▶ Observers for Nonlinear Systems
▶ Observers in Linear Systems Theory
▶ Particle Filters

Bibliography

Abur A, Gómez Espósito A (2004) Power system state estimation – theory and implementation. Marcel Dekker, New York

Aidala VJ, Hammel SE (1983) Utilization of modified polar coordinates for bearings-only tracking. IEEE Trans Autom Control 28(3):283–294

Alamo T, Bravo JM, Camacho EF (2005) Guaranteed state estimation by zonotopes. Automatica 41(6): 1035–1043

Alessandri A, Baglietto M, Battistelli G (2005) Receding-horizon estimation for switching discrete-time linear systems. IEEE Trans Autom Control 50(11):1736–1748

Alessandri A, Baglietto M, Battistelli G (2008) Moving-horizon state estimation for nonlinear discrete-time systems: new stability results and approximation schemes. Automatica 44(7):1753–1765

Alriksson P, Rantzer A (2006) Distributed Kalman filtering using weighted averaging. In: Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto

Alspach DL, Sorenson HW (1972) Nonlinear Bayesian estimation using Gaussian sum approximations. IEEE Trans Autom Control 17(4):439–448

Anderson BDO, Chirarattananon S (1972) New linear smoothing formulas. IEEE Trans Autom Control 17(1):160–161

Anderson BDO, Moore JB (1979) Optimal filtering. Prentice Hall, Englewood Cliffs

Andrews A (1968) A square root formulation of the Kalman covariance equations. AIAA J 6(6):1165–1166

Andrews HC (1972) Introduction to mathematical techniques in pattern recognition. Wiley, New York

Aoki M (1987) State space modeling of time-series. Springer, Berlin

Athans M (1971) An optimal allocation and guidance laws for linear interception and rendezvous problems. IEEE Trans Aerosp Electron Syst 7(5):843–853

Balakrishnan AV, Lions JL (1967) State estimation for infinite-dimensional systems. J Comput Syst Sci 1(4):391–403

Barbarisi O, Vasca F, Glielmo L (2006) State of charge Kalman filter estimator for automotive batteries. Control Eng Pract 14(3):267–275

Bard Y (1974) Nonlinear parameter estimation. Academic, New York

Barkat M (2005) Signal detection and estimation. Artech House, Boston

Bar-Shalom Y, Fortmann TE (1988) Tracking and data association. Academic, Boston

Bar-Shalom Y, Li X, Kirubarajan T (2001) Estimation with applications to tracking and navigation. Wiley, New York

Bar-Shalom Y, Willett P, Tian X (2013) Tracking and data fusion: a handbook of algorithms. YBS Publishing Storrs, CT

Battistelli G, Chisci L, Morrocchi S, Papi F (2011) An information-theoretic approach to distributed state estimation. In: Proceedings of the 18th IFAC World Congress, Milan, pp 12477–12482

Battistelli G, Chisci L, Mugnai G, Farina A, Graziano A (2012) Consensus-based algorithms for distributed filtering. In: Proceedings of the 51st IEEE Control and Decision Conference, Maui, pp 794–799

Battistelli G, Chisci L (2014) Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. Automatica 50:707–718

Battistelli G, Chisci L, Fantacci C, Farina A, Graziano A (2013) Consensus CPHD filter for distributed multitarget tracking. IEEE J Sel Topics Signal Process 7(3):508–520

Bayes T (1763) Essay towards solving a problem in the doctrine of chances. Philos Trans R Soc Lond 53: 370–418. doi:10.1098/rstl.1763.0053

Bayless JW, Brigham EO (1970) Application of the Kalman filter. Geophysics 35(1):2–23

Bekey GA (1973) Parameter estimation in biological systems: a survey. In: Proceedings of the 3rd IFAC Symposium Identification and System Parameter Estimation, Delft, pp 1123–1130

Benedict TR, Bordner GW (1962) Synthesis of an optimal set of radar track-while-scan smoothing equations. IRE Trans Autom Control 7(4):27–32

Benes VE (1981) Exact finite-dimensional filters for certain diffusions with non linear drift. Stochastics 5(1):65–92

Bertsekas DP, Rhodes IB (1971) Recursive state estimation for a set-membership description of uncertainty. IEEE Trans Autom Control 16(2):117–128

Biemond J, Rieske J, Gerbrands JJ (1983) A fast Kalman filter for images degraded by both blur and noise. IEEE Trans Acoust Speech Signal Process 31(5):1248–1256

Bierman GJ (1974) Sequential square root filtering and smoothing of discrete linear systems. Automatica 10(2):147–158

Bierman GJ (1977) Factorization methods for discrete sequential estimation. Academic, New York

Blackman S, Popoli R (1999) Design and analysis of modern tracking systems. Artech House, Norwood

Blasch EP, Lambert DA, Valin P, Kokar MM, Llinas J, Das S, Chong C, Shahbazian E (2012) High level information fusion (HLIF): survey of models, issues, and grand challenges. IEEE Aerosp Electron Syst Mag 27(9):4–20

Bryson AE, Frazier M (1963) Smoothing for linear and nonlinear systems, TDR 63-119, Aero System Division. Wright-Patterson Air Force Base, Ohio

Calafiore GC, Abrate F (2009) Distributed linear estimation over sensor networks. Int J Control 82(5):868–882

Carravetta F, Germani A, Raimondi M (1996) Polynomial filtering for linear discrete-time non-Gaussian systems. SIAM J Control Optim 34(5):1666–1690

Chen J, Patton RJ (1999) Robust model-based fault diagnosis for dynamic systems. Kluwer, Boston

Chisci L, Zappa G (1992) Square-root Kalman filtering of descriptor systems. Syst Control Lett 19(4):325–334

Chisci L, Garulli A, Zappa G (1996) Recursive state bounding by parallelotopes. Automatica 32(7): 1049–1055

Chou KC, Willsky AS, Benveniste A (1994) Multiscale recursive estimation, data fusion, and regularization. IEEE Trans Autom Control 39(3):464–478

Combettes P (1993) The foundations of set-theoretic estimation. Proc IEEE 81(2):182–208

Cramér H (1946) Mathematical methods of statistics. University Press, Princeton

Crisan D, Rozovskii B (eds) (2011) The Oxford handbook of nonlinear filtering. Oxford University Press, Oxford

Dai L (1987) State estimation schemes in singular systems. In: Preprints 10th IFAC World Congress, Munich, vol 9, pp 211–215

Dai L (1989) Filtering and LQG problems for discrete-time stochastic singular systems. IEEE Trans Autom Control 34(10):1105–1108

E

Darragh J, Looze D (1984) Noncausal minimax linear state estimation for systems with uncertain second-order statistics. IEEE Trans Autom Control 29(6):555–557

Daum FE (1986) Exact finite dimensional nonlinear filters. IEEE Trans Autom Control 31(7):616–622

Daum F (2005) Nonlinear filters: beyond the Kalman filter. IEEE Aerosp Electron Syst Mag 20(8):57–69

Debs AS, Larson RE (1970) A dynamic estimator for tracking the state of a power system. IEEE Trans Power Appar Syst 89(7):1670–1678

Devlin K (2008) The unfinished game: Pascal, Fermat and the seventeenth-century letter that made the world modern. Basic Books, New York. Copyright (C) Keith Devlin

Dissanayake G, Newman P, Clark S, Durrant-Whyte H, Csorba M (2001) A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans Robot Autom 17(3):229–241

Dochain D, Vanrolleghem P (2001) Dynamical modelling and estimation of wastewater treatment processes. IWA Publishing, London

Doucet A, de Freitas N, Gordon N (eds) (2001) Sequential Monte Carlo methods in practice. Springer, New York

Duncan TE (1970) On the calculation of mutual information. SIAM J Appl Math 19(1):215–220

Durrant-Whyte H, Bailey T (2006a) Simultaneous localization and mapping: Part I. IEEE Robot Autom Mag 13(2):99–108

Durrant-Whyte H, Bailey T (2006b) Simultaneous localization and mapping: Part II. IEEE Robot Autom Mag 13(3):110–117

Elliott RJ, Aggoun L, Moore JB (2008) Hidden Markov models: estimation and control. Springer, New York. (first edition in 1995)

Evensen G (1994a) Inverse methods and data assimilation in nonlinear ocean models. Physica D 77:108–129

Evensen G (1994b) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. J Geophys Res 99(10):143–162

Evensen G (2007) Data assimilation – the ensemble Kalman filter. Springer, Berlin

Falb PL (1967) Infinite-dimensional filtering; the Kalman-Bucy filter in Hilbert space. Inf Control 11(1):102–137

Farina A (1999) Target tracking with bearings-only measurements. Signal Process 78(1):61–78

Farina A, Studer FA (1985) Radar data processing. Volume 1 – introduction and tracking, Editor P. Bowron. Research Studies Press, Letchworth/Wiley, New York (Translated in Russian (Radio I Sviaz, Moscow 1993) and in Chinese. China Defense Publishing House, 1988)

Farina A, Studer FA (1986) Radar data processing. Volume 2 – advanced topics and applications, Editor P. Bowron. Research Studies Press, Letchworth/Wiley, New York (In Chinese, China Defense Publishing House, 1992)

Farrell JA, Barth M (1999) The global positioning system and inertial navigation. McGraw-Hill, New York

Fisher RA (1912) On an absolute criterion for fitting frequency curves. Messenger Math 41(5):155–160

Fisher RA (1922) On the mathematical foundations of theoretical statistics. Philos Trans R Soc Lond Ser A 222:309–368

Fisher RA (1925) Theory of statistical estimation. Math Proc Camb Philos Soc 22(5):700–725

Flinn EA, Robinson EA, Treitel S (1967) Special issue on the MIT geophysical analysis group reports. Geophysics 32:411–525

Frost PA, Kailath T (1971) An innovations approach to least-squares estimation, Part III: nonlinear estimation in white Gaussian noise. IEEE Trans Autom Control 16(3):217–226

Galilei G (1632) Dialogue concerning the two chief world systems (Translated in English by S. Drake (Foreword of A. Einstein), University of California Press, Berkeley, 1953)

Gauss CF (1806) Theoria motus corporum coelestium in sectionibus conicis solem ambientum (Translated in English by C.H. Davis, Reprinted as Theory of motion of the heavenly bodies, Dover, New York, 1963)

Gauss CF (1995) Theoria combinationis observationum erroribus minimis obnoxiae (Translated by G.W. Stewart, Classics in Applied Mathematics), vol 11. SIAM, Philadelphia

Germani A, Manes C, Palumbo P (2005) Polynomial extended Kalman filter. IEEE Trans Autom Control 50(12):2059–2064

Ghil M, Malanotte-Rizzoli P (1991) Data assimilation in meteorology and oceanography. Adv Geophys 33:141–265

Ghosh M, Mukhopadhyay N, Sen PK (1997) Sequential estimation. Wiley, New York

Golub GH (1965) Numerical methods for solving linear least squares problems. Numerische Mathematik 7(3):206–216

Goodwin GC, Seron MM, De Doná JA (2005) Constrained control and estimation. Springer, London

Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non Gaussian Bayesian state estimation. IEE Proc F 140(2):107–113

Grewal MS, Weill LR, Andrews AP (2001) Global positioning systems, inertial navigation and integration. Wiley, New York

Grimble MJ (1988) H_∞ design of optimal linear filters. In: Byrnes C, Martin C, Saeks R (eds) Linear circuits, systems and signal processing: theory and applications. North Holland, Amsterdam, pp 535–540

Guo D, Shamai S, Verdú S (2005) Mutual information and minimum mean-square error in Gaussian channels. IEEE Trans Inf Theory 51(4):1261–1282

Hall DL, Llinas J (1997) An introduction to multisensor data fusion. Proc IEEE 85(1):6–23

Hassibi B, Sayed AH, Kailath T (1999) Indefinite-quadratic estimation and control. SIAM, Philadelphia

Heemink AW, Segers AJ (2002) Modeling and prediction of environmental data in space and time using Kalman filtering. Stoch Environ Res Risk Assess 16: 225–240

Helstrom CW (1969) Quantum detection and estimation theory. J Stat Phys 1(2):231–252

Helstrom CW (1976) Quantum detection and estimation theory. Academic, New York

Hernandez M, Farina A, Ristic B (2006) PCRLB for tracking in cluttered environments: measurement sequence conditioning approach. IEEE Trans Aerosp Electron Syst 42(2):680–704

Ho YC, Agrawala AK (1968) On pattern classification algorithms – introduction and survey. Proc IEEE 56(12):2101–2113

Ho YC, Lee RCK (1964) A Bayesian approach to problems in stochastic estimation and control. IEEE Trans Autom Control 9(4):333–339

Iida S, Ohki F, Yamamoto N (2010) Robust quantum Kalman filtering under the phase uncertainty of the probe-laser. In: IEEE international symposium on computer-aided control system design, Yokohama, pp 749–754

Itakura F (1971) Extraction of feature parameters of speech by statistical methods. In: Proceedings of the 8th Symposium Speech Information Processing, Sendai, pp II-5.1–II-5.12

Jazwinski AH (1966) Filtering for nonlinear dynamical systems. IEEE Trans Autom Control 11(4):765–766

Jazwinski AH (1968) Limited memory optimal filtering. IEEE Trans Autom Control 13(5): 558–563

Jazwinski AH (1970) Stochastic processes and filtering theory. Academic, New York

Joseph DP, Tou TJ (1961) On linear control theory. Trans Am Inst Electr Eng 80(18):193–196

Julier SJ, Uhlmann JK (1997) A non-divergent estimation algorithm in the presence of unknown correlations. In: Proceedings of the 1997 American Control Conference, Albuquerque, vol 4, pp 2369–2373

Julier SJ, Uhlmann JK (2004) Unscented Filtering and Nonlinear Estimation. Proc IEEE 92(3): 401–422

Julier SJ, Uhlmann JK, Durrant-Whyte H (1995) A new approach for filtering nonlinear systems. In: Proceedings of the 1995 American control conference, Seattle, vol 3, pp 1628–1632

Kailath T (1968) An innovations approach to least-squares estimation, Part I: linear filtering in additive white noise. IEEE Trans Autom Control 13(6): 646–655

Kailath T (1970) The innovations approach to detection and estimation theory. Proc IEEE 58(5): 680–695

Kailath T (1973) Some new algorithms for recursive estimation in constant linear systems. IEEE Trans Inf Theory 19(6):750–760

Kailath T, Frost PA (1968) An innovations approach to least-squares estimation, Part II: linear smoothing in additive white noise. IEEE Trans Autom Control 13(6):655–660

Kailath T, Sayed AH, Hassibi B (2000) Linear estimation. Prentice Hall, Upper Saddle River

Kalman RE (1960a) On the general theory of control systems. In: Proceedings of the 1st International Congress of IFAC, Moscow, USSR, vol 1, pp 481–492

Kalman RE (1960b) A new approach to linear filtering and prediction problems. Trans ASME J Basic Eng Ser D 82(1):35–45

Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory. Trans ASME J Basic Eng Ser D 83(3):95–108

Kamgarpour M, Tomlin C (2007) Convergence properties of a decentralized Kalman filter. In: Proceedings of the 47th IEEE Conference on Decision and Control, New Orleans, pp 5492–5498

Kaminski PG, Bryson AE (1972) Discrete square-root smoothing. In: Proceedings of the AIAA Guidance and Control Conference, paper no. 72–877, Stanford

Kay SM (1993) Fundamentals of statistical signal processing, volume 1: estimation theory. Prentice Hall, Englewood Cliffs

Kim K, Shevlyakov G (2008) Why Gaussianity? IEEE Signal Process Mag 25(2):102–113

Kim J, Woods JW (1998) 3-D Kalman filter for image motion estimation. IEEE Trans Image Process 7(1):42–52

Koch KR (1999) Parameter estimation and hypothesis testing in linear models. Springer, Berlin

Kolmogorov AN (1933) Grundbegriffe der Wahrscheinlichkeitsrechnung (in German) (English translation edited by Nathan Morrison: Foundations of the theory of probability, Chelsea, New York, 1956). Springer, Berlin

Kolmogorov AN (1941) Interpolation and extrapolation of stationary random sequences. Izv Akad Nauk SSSR Ser Mat 5:3–14 (in Russian) (English translation by G. Lindqvist, published in Selected works of A.N. Kolmogorov - Volume II: Probability theory and mathematical statistics, A.N. Shyryaev (Ed.), pp. 272–280, Kluwer, Dordrecht, Netherlands, 1992)

Kushner HJ (1962) On the differential equations satisfied by conditional probability densities of Markov processes with applications. SIAM J Control Ser A 2(1):106–199

Kushner HJ (1967) Dynamical equations for optimal nonlinear filtering. J Differ Equ 3(2):179–190

Kushner HJ (1970) Filtering for linear distributed parameter systems. SIAM J Control 8(3): 346–359

Kwakernaak H (1967) Optimal filtering in linear systems with time delays. IEEE Trans Autom Control 12(2):169–173

Lainiotis DG (1972) Adaptive pattern recognition: a state-variable approach. In: Watanabe S (ed) Frontiers of pattern recognition. Academic, New York

Lainiotis DG (1974) Estimation: a brief survey. Inf Sci 7:191–202

Laplace PS (1814) Essai philosophique sur le probabilités (Translated in English by F.W. Truscott, F.L. Emory, Wiley, New York, 1902)

Legendre AM (1810) Méthode des moindres quarrés, pour trouver le milieu le plus probable entre les résultats de différentes observations. In: Mémoires de la Classe des Sciences Mathématiques et Physiques de l'Institut Impérial de France, pp 149–154. Originally appeared as appendix to the book Nouvelle méthodes pour la détermination des orbites des comètes, 1805

E

Lehmann EL, Casella G (1998) Theory of point estimation. Springer, New York

Leibungudt BG, Rault A, Gendreau F (1983) Application of Kalman filtering to demographic models. IEEE Trans Autom Control 28(3):427–434

Levy BC (2008) Principles of signal detection and parameter estimation. Springer, New York

Ljung L (1999) System identification: theory for the user. Prentice Hall, Upper Saddle River

Los Alamos Scientific Laboratory (1966) Fermi invention rediscovered at LASL. The Atom, pp 7–11

Luenberger DG (1964) Observing the state of a linear system. IEEE Trans Mil Electron 8(2):74–80

Mahler RPS (1996) A unified foundation for data fusion. In: Sadjadi FA (ed) Selected papers on sensor and data fusion. SPIE MS-124. SPIE Optical Engineering Press, Bellingham, pp 325–345. Reprinted from 7th Joint Service Data Fusion Symposium, pp 154–174, Laurel (1994)

Mahler RPS (2003) Multitarget Bayes filtering via first-order multitarget moments. IEEE Trans Aerosp Electron Syst 39(4):1152–1178

Mahler RPS (2007a) Statistical multisource multitarget information fusion. Artech House, Boston

Mahler RPS (2007b) PHD filters of higher order in target number. IEEE Trans Aerosp Electron Syst 43(4):1523–1543

Mangoubi ES (1998) Robust estimation and failure detection: a concise treatment. Springer, New York

Maybeck PS (1979 and 1982) Stochastic models, estimation and control, vols I, II and III. Academic, New York. 1979 (volume I) and 1982 (volumes II and III)

Mayer-Wolf E, Zakai M (1983) On a formula relating the Shannon information to the Fisher information for the filtering problem. In: Korezlioglu H, Mazziotto G and Szpirglas J (eds) Lecture notes in control and information sciences, vol 61. Springer, New York, pp 164–171

Mayne DQ (1966) A solution of the smoothing problem for linear dynamic systems. Automatica 4(2):73–92

McGarty TP (1971) The estimation of the constituent densities of the upper atmosphere. IEEE Trans Autom Control 16(6):817–823

McGee LA, Schmidt SF (1985) Discovery of the Kalman filter as a practical tool for aerospace and industry, NASA-TM-86847

McGeoch CC, Wang C (2013) Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In: Computing Frontiers 2013 Conference, Ischia, article no. 23. doi:10.1145/2482767.2482797

McGrayne SB (2011) The theory that would not die. How Bayes' rule cracked the enigma code, hunted down Russian submarines, and emerged triumphant from two centuries of controversies. Yale University Press, New Haven/London

Meditch JS (1967) Orthogonal projection and discrete optimal linear smoothing. SIAM J Control 5(1):74–89

Meditch JS (1971) Least-squares filtering and smoothing for linear distributed parameter systems. Automatica 7(3):315–322

Mendel JM (1977) White noise estimators for seismic data processing in oil exploration. IEEE Trans Autom Control 22(5):694–706

Mendel JM (1983) Optimal seismic deconvolution: an estimation-based approach. Academic, New York

Mendel JM (1990) Maximum likelihood deconvolution – a journey into model-based signal processing. Springer, New York

Metropolis N, Ulam S (1949) The Monte Carlo method. J Am Stat Assoc 44(247):335–341

Milanese M, Belforte G (1982) Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors: linear families of models and estimators. IEEE Trans Autom Control 27(2):408–414

Milanese M, Vicino A (1993) Optimal estimation theory for dynamic systems with set-membership uncertainty. Automatica 27(6):427–446

Miller WL, Lewis JB (1971) Dynamic state estimation in power systems. IEEE Trans Autom Control 16(6):841–846

Mlodinow L (2009) The drunkard's walk: how randomness rules our lives. Pantheon Books, New York. Copyright (C) Leonard Mlodinow

Monticelli A (1999) State estimation in electric power systems: a generalized approach. Kluwer, Dordrecht

Morf M, Kailath T (1975) Square-root algorithms for least-squares estimation. IEEE Trans Autom Control 20(4):487–497

Morf M, Sidhu GS, Kailath T (1974) Some new algorithms for recursive estimation in constant, linear, discrete-time systems. IEEE Trans Autom Control 19(4):315–323

Mullane J, Vo B-N, Adams M, Vo B-T (2011) Random finite sets for robotic mapping and SLAM. Springer, Berlin

Nachazel K (1993) Estimation theory in hydrology and water systems. Elsevier, Amsterdam

Najim M (2008) Modeling, estimation and optimal filtering in signal processing. Wiley, Hoboken. First published in France by Hermes Science/Lavoisier as Modélisation, estimation et filtrage optimal en traitement du signal (2006)

Neal SR (1967) Discussion of parametric relations for the filter predictor. IEEE Trans Autom Control 12(3):315–317

Netz R, Noel W (2011) The Archimedes codex. Phoenix, Philadelphia, PA

Nikoukhah R, Willsky AS, Levy BC (1992) Kalman filtering and Riccati equations for descriptor systems. IEEE Trans Autom Control 37(9):1325–1342

Olfati-Saber R (2007) Distributed Kalman filtering for sensor networks. In: Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, pp 5492–5498

Olfati-Saber R, Fax JA, Murray R (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

Painter JH, Kerstetter D, Jowers S (1990) Reconciling steady-state Kalman and $\alpha - \beta$ filter design. IEEE Trans Aerosp Electron Syst 26(6):986–991

Park S, Serpedin E, Qarage K (2013) Gaussian assumption: the least favorable but the most useful. IEEE Signal Process Mag 30(3):183–186

Personick SD (1971) Application of quantum estimation theory to analog communication over quantum channels. IEEE Trans Inf Theory 17(3):240–246

Pindyck RS, Roberts SM (1974) Optimal policies for monetary control. Ann Econ Soc Meas 3(1): 207–238

Poor HV (1994) An introduction to signal detection and estimation, 2nd edn. Springer, New York. (first edition in 1988)

Poor HV, Looze D (1981) Minimax state estimation for linear stochastic systems with noise uncertainty. IEEE Trans Autom Control 26(4):902–906

Potter JE, Stern RG (1963) Statistical filtering of space navigation measurements. In: Proceedings of the 1963 AIAA Guidance and Control Conference, paper no. 63–333, Cambridge

Ramm AG (2005) Random fields estimation. World Scientific, Hackensack

Rao C (1945) Information and the accuracy attainable in the estimation of statistical parameters. Bull Calcutta Math Soc 37:81–89

Rao CV, Rawlings JB, Lee JH (2001) Constrained linear state estimation – a moving horizon approach. Automatica 37(10):1619–1628

Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. IEEE Trans Autom Control 48(2): 246–257

Rauch HE (1963) Solutions to the linear smoothing problem. IEEE Trans Autom Control 8(4): 371–372

Rauch HE, Tung F, Striebel CT (1965) Maximum likelihood estimates of linear dynamic systems. AIAA J 3(8):1445–1450

Reid D (1979) An algorithm for tracking multiple targets. IEEE Trans Autom Control 24(6):843–854

Riccati JF (1722) Animadversiones in aequationes differentiales secundi gradi. Actorum Eruditorum Lipsiae, Supplementa 8:66–73. Observations regarding differential equations of second order, translation of the original Latin into English, by I. Bruce, 2007

Riccati JF (1723) Appendix in animadversiones in aequationes differentialis secundi gradi. Acta Eruditorum Lipsiae

Ristic B (2013) Particle filters for random set models. Springer, New York

Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter: particle filters for tracking applications. Artech House, Boston

Ristic B, Vo B-T, Vo B-N, Farina A (2013) A tutorial on Bernoulli filters: theory, implementation and applications. IEEE Trans Signal Process 61(13):3406–3430

Robinson EA (1963) Mathematical development of discrete filters for detection of nuclear explosions. J Geophys Res 68(19):5559–5567

Roman WS, Hsu C, Habegger LT (1971) Parameter identification in a nonlinear reactor system. IEEE Trans Nucl Sci 18(1):426–429

Sage AP, Masters GW (1967) Identification and modeling of states and parameters of nuclear reactor systems. IEEE Trans Nucl Sci 14(1):279–285

Sage AP, Melsa JL (1971) Estimation theory with applications to communications and control. McGraw-Hill, New York

Schmidt SF (1966) Application of state-space methods to navigation problems. Adv Control Syst 3:293–340

Schmidt SF (1970) Computational techniques in Kalman filtering, NATO-AGARD-139

Schonhoff TA, Giordano AA (2006) Detection and estimation theory and its applications. Pearson Prentice Hall, Upper Saddle River

Schweppe FC (1968) Recursive state estimation with unknown but bounded errors and system inputs. IEEE Trans Autom Control 13(1):22–28

Segall A (1976) Recursive estimation from discrete-time point processes. IEEE Trans Inf Theory 22(4): 422–431

Servi L, Ho Y (1981) Recursive estimation in the presence of uniformly distributed measurement noise. IEEE Trans Autom Control 26(2):563–565

Simon D (2006) Optimal state estimation – Kalman, H_∞ and nonlinear approaches. Wiley, New York

Simpson HR (1963) Performance measures and optimization condition for a third-order tracker. IRE Trans Autom Control 8(2):182–183

Sklansky J (1957) Optimizing the dynamic parameters of a track-while-scan system. RCA Rev 18:163–185

Smeth P, Ristic B (2004) Kalman filter and joint tracking and classification in TBM framework. In: Proceedings of the 7th international conference on information fusion, Stockholm, pp 46–53

Smith R, Self M, Cheeseman P (1986) Estimating uncertain spatial relationships in robotics. In: Proceedings of the 2nd conference on uncertainty in artificial intelligence, Philadelphia, pp 435–461

Snijders TAB, Koskinen J, Schweinberger M (2012) Maximum likelihood estimation for social network dynamics. Ann Appl Stat 4(2):567–588

Snyder DL (1968) The state-variable approach to continuous estimation with applications to analog communications. MIT, Cambridge

Snyder DL (1970) Estimation of stochastic intensity functions of conditional Poisson processes. Monograph no. 128, Biomedical Computer Laboratory, Washington University, St. Louis

Söderström T (1994) Discrete-time stochastic system – estimation & control. Prentice Hall, New York

Söderström T, Stoica P (1989) System identification. Prentice Hall, New York

Sorenson HW, Alspach DL (1970) Gaussian sum approximations for nonlinear filtering. In: Proceedings of the 1970 IEEE Symposium on Adaptive Processes, Austin, TX pp 19.3.1–19.3.9

Sorenson HW, Alspach DL (1971) Recursive Bayesian estimation using Gaussian sums. Automatica 7(4): 465–479

Stankovic SS, Stankovic MS, Stipanovic DM (2009) Consensus based overlapping decentralized estimation

E

with missing observations and communication faults. Automatica 45(6):1397–1406

Stark L (1968) Neurological control systems studies in bioengineering. Plenum Press, New York

Stengel PF (1994) Optimal control and estimation. Dover, New York. Originally published as Stochastic optimal control. Wiley, New York, 1986

Stephant J, Charara A, Meizel D (2004) Virtual sensor: application to vehicle sideslip angle and transversal forces. IEEE Trans Ind Electron 51(2):278–289

Stratonovich RL (1959) On the theory of optimal nonlinear filtering of random functions. Theory Probab Appl 4:223–225

Stratonovich RL (1960) Conditional Markov processes. Theory Probab Appl 5(2):156–178

Taylor JH (1979) The Cramér-Rao estimation error lower bound computation for deterministic nonlinear systems. IEEE Trans Autom Control 24(2): 343–344

Thrun S, Burgard W, Fox D (2006) Probabilistic robotics. MIT, Cambridge

Tichavsky P, Muravchik CH, Nehorai A (1998) Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. IEEE Trans Signal Process 6(5):1386–1396

Toyoda J, Chen MS, Inoue Y (1970) An application of state estimation to short-term load forecasting, Part I: forecasting model and Part II: implementation. IEEE Trans Power Appar Syst 89(7):1678–1682 (Part I) and 1683–1688 (Part II)

Tsybakov AB (2009) Introduction to nonparametric estimation. Springer, New York

Tuncer TE, Friedlander B (2009) Classical and modern direction-of-arrival estimation. Academic, Burlington

Tzafestas SG, Nightingale JM (1968) Optimal filtering, smoothing and prediction in linear distributed-parameter systems. Proc Inst Electr Eng 115(8): 1207–1212

Ulam S (1952) Random processes and transformations. In: Proceedings of the International Congress of Mathematicians, Cambridge, vol 2, pp 264–275

Ulam S, Richtmeyer RD, von Neumann J (1947) Statistical methods in neutron diffusion, Los Alamos Scientific Laboratory report LAMS-551

Van Trees H (1971) Detection, estimation and modulation theory – Parts I, II and III. Wiley, New York. Republished in 2013

van Trees HL, Bell KL (2007) Bayesian bounds for parameter estimation and nonlinear filtering/tracking. Wiley, Hoboken/IEEE, Piscataway

Venerus JC, Bullock TE (1970) Estimation of the dynamic reactivity using digital Kalman filtering. Nucl Sci Eng 40:199–205

Verdú S, Poor HV (1984) Minimax linear observers and regulators for stochastic systems with uncertain second-order statistics. IEEE Trans Autom Control 29(6):499–511

Vicino A, Zappa G (1996) Sequential approximation of feasible parameter sets for identification with set membership uncertainty. IEEE Trans Autom Control 41(6):774–785

Vo B-N, Ma WK (1996) The Gaussian mixture probability hypothesis density filter. IEEE Trans Signal Process 54(11):4091–4101

Vo B-T, Vo B-N, Cantoni A (2007) Analytic implementations of the cardinalized probability hypothesis density filter. IEEE Trans Signal Process 55(7):3553–3567

Wakita H (1973) Estimation of the vocal tract shape by inverse optimal filtering. IEEE Trans Audio Electroacoust 21(5):417–427

Wertz W (1978) Statistical density estimation – a survey. Vandenhoeck and Ruprecht, Göttingen

Wiener N (1949) Extrapolation, interpolation and smoothing of stationary time-series. MIT, Cambridge

Willsky AS (1976) A survey of design methods for failure detection in dynamic systems. Automatica 12(6): 601–611

Wonham WM (1965) Some applications of stochastic differential equations to optimal nonlinear filtering. SIAM J Control 2(3):347–369

Woods JW, Radewan C (1977) Kalman filtering in two dimensions. IEEE Trans Inf Theory 23(4): 473–482

Xiao L, Boyd S, Lall S (2005) A scheme for robust distributed sensor fusion based on average consensus. In: Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, pp 63–70

Zachrisson LE (1969) On optimal smoothing of continuous-time Kalman processes. Inf Sci 1(2): 143–172

Zellner A (1971) An introduction to Bayesian inference in econometrics. Wiley, New York

Event-Triggered and Self-Triggered Control

W.P.M.H. Heemels[1], Karl H. Johansson[2], and Paulo Tabuada[3]

[1]Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

[2]ACCESS Linnaeus Center, Royal Institute of Technology, Stockholm, Sweden

[3]Department of Electrical Engineering, University of California, Los Angeles, CA, USA

Abstract

Recent developments in computer and communication technologies have led to a new type of large-scale resource-constrained wireless

embedded control systems. It is desirable in these systems to limit the sensor and control computation and/or communication to instances when the system needs attention. However, classical sampled-data control is based on performing sensing and actuation periodically rather than when the system needs attention. This article discusses event- and self-triggered control systems where sensing and actuation is performed when needed. Event-triggered control is reactive and generates sensor sampling and control actuation when, for instance, the plant state deviates more than a certain threshold from a desired value. Self-triggered control, on the other hand, is proactive and computes the next sampling or actuation instance ahead of time. The basics of these control strategies are introduced together with references for further reading.

Keywords

Event-triggered control; Hybrid systems; Real-time control; Resource-constrained embedded control; Sampled-data systems; Self-triggered control

Introduction

In standard control textbooks, e.g., Aström and Wittenmark (1997) and Franklin et al. (2010), periodic control is presented as the only choice for implementing feedback control laws on digital platforms. Although this time-triggered control paradigm has proven to be extremely successful in many digital control applications, recent developments in computer and communication technologies have led to a new type of large-scale resource-constrained (wireless) control systems that call for a reconsideration of this traditional paradigm. In particular, the increasing popularity of (shared) wired and wireless networked control systems raises the importance of explicitly addressing energy, computation, and communication constraints when designing feedback control loops. Aperiodic control strategies that allow the inter-execution times of control tasks to be varying in time offer potential advantages with

respect to periodic control when handling these constraints, but they also introduce many new interesting theoretical and practical challenges.

Although the discussions regarding periodic vs. aperiodic implementation of feedback control loops date back to the beginning of computer-controlled systems, e.g., Gupta (1963), in the late 1990s two influential papers (Årzén 1999; Åström and Bernhardsson 1999) highlighted the advantages of *event-based* feedback control. These two papers spurred the development of the first *systematic designs* of event-based implementations of stabilizing feedback control laws, e.g., Yook et al. (2002), Tabuada (2007), Heemels et al. (2008), and Henningsson et al. (2008). Since then, several researchers have improved and generalized these results and alternative approaches have appeared. In the meantime, also so-called *self-triggered* control (Velasco et al. 2003) emerged. Event-triggered and self-triggered control systems consist of two elements, namely, a feedback controller that computes the control input and a triggering mechanism that determines when the control input has to be updated again. The difference between event-triggered control and self-triggered control is that the former is reactive, while the latter is proactive. Indeed, in event-triggered control, a triggering condition based on current measurements is continuously monitored and when the condition holds, an event is triggered. In self-triggered control the next update time is precomputed at a control update time based on predictions using previously received data and knowledge of the plant dynamics. In some cases, it is advantageous to combine event-triggered and self-triggered control resulting in a control system reactive to unpredictable disturbances and proactive by predicting future use of resources.

Time-Triggered, Event-Triggered and Self-Triggered Control

To indicate the differences between various digital implementations of feedback control laws, consider the control of the nonlinear plant

$$\dot{x} = f(x, u) \qquad (1)$$

with $x \in \mathbb{R}^{n_x}$ the state variable and $u \in \mathbb{R}^{n_u}$ the input variable. The system is controlled by a nonlinear state feedback law

$$u = h(x) \qquad (2)$$

where $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ is an appropriate mapping that has to be implemented on a digital platform. Recomputing the control value and updating the actuator signals will occur at times denoted by $t_0, t_1, t_2, \dots$ with $t_0 = 0$. If we assume the inputs to be held constant in between the successive recomputations of the control law (referred to as sample-and-hold or zero-order-hold), we have

$$u(t) = u(t_k) = h(x(t_k)) \quad \forall t \in [t_k, t_{k+1}), \ k \in \mathbb{N}. \qquad (3)$$

We refer to the instants $\{t_k\}_{k \in \mathbb{N}}$ as the *triggering* times or *execution* times. Based on these times we can easily explain the difference between time-triggered control, event-triggered control, and self-triggered control.

In *time-triggered* control we have the equality $t_k = kT_s$ with $T_s > 0$ being the sampling period. Hence, the updates take place equidistantly in time irrespective of how the system behaves. There is no "feedback mechanism" in determining the execution times; they are determined a priori and in "open loop." Another way of writing the triggering mechanism in time-triggered control is

$$t_{k+1} = t_k + T_s, \ k \in \mathbb{N} \qquad (4)$$

with $t_0 = 0$.

In *event-triggered* control the next execution time of the controller is determined by an event-triggering mechanism that continuously verifies if a certain condition based on the actual state variable becomes true. This condition includes often also information on the state variable $x(t_k)$ at the previous execution time t_k and can be written, for instance, as $C(x(t), x(t_k)) > 0$. Formally, the execution times are then determined by

$$t_{k+1} = \inf\{t > t_k \mid C(x(t), x(t_k)) > 0\} \qquad (5)$$

with $t_0 = 0$. Hence, it is clear from (5) that there is a *feedback* mechanism present in the determination of the next execution time as it is based on the measured state variable. In this sense event-triggered control is *reactive*.

Finally, in *self-triggered* control the next execution time is determined *proactively* based on the measured state $x(t_k)$ at the previous execution time. In particular, there is a function $M : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ that specifies the next execution time as

$$t_{k+1} = t_k + M(x(t_k)) \qquad (6)$$

with $t_0 = 0$. As a consequence, in self-triggered control both the control value $u(t_k)$ and the next execution time t_{k+1} are computed at execution time t_k. In between t_k and t_{k+1}, no further actions are required from the controller. Note that the time-triggered implementation can be seen as a special case of the self-triggered implementation by taking $M(x) = T_s$ for all $x \in \mathbb{R}^{n_x}$.

Clearly, in all the three implementation schemes T_s, C and M are chosen together with the feedback law given through h to provide stability and performance guarantees and to realize a certain utilization of computer and communication resources.

Lyapunov-Based Analysis

Much work on event-triggered control used one of the following two modeling and analysis frameworks: The perturbation approach and the hybrid system approach.

Perturbation Approach

In the perturbation approach one adopts perturbed models that describe how the event-triggered implementation of the control law perturbs the ideal continuous-time implementation $u(t) = h(x(t))$, $t \in \mathbb{R}_{\geq 0}$. In order to do so, consider the error e given by

$$e(t) = x(t_k) - x(t) \ \text{for } t \in [t_k, t_{k+1}), \ k \in \mathbb{N}. \qquad (7)$$

Using this error variable we can write the closed-loop system based on (1) and (3) as

$$\dot{x} = f(x, h(x + e)). \qquad (8)$$

Essentially, the three implementations discussed above have their own way of indicating when an execution takes place and the error e is reset to zero. The equation (8) clearly shows how the ideal closed-loop system is *perturbed* by using a time-triggered, event-triggered, or self-triggered implementation of the feedback law in (2). Indeed, when $e = 0$ we obtain the ideal closed loop

$$\dot{x} = f(x, h(x)). \qquad (9)$$

The control law in (2) is typically chosen so as to guarantee that the system in (9) has certain global asymptotic stability (GAS) properties. In particular, it is often assumed that there exists a Lyapunov function $V : \mathbb{R}_{n_x} \to \mathbb{R}_{\geq 0}$ in the sense that V is positive definite and for all $x \in \mathbb{R}^{n_x}$ we have

$$\frac{\partial V}{\partial x} f(x, h(x)) \leq -\|x\|^2. \qquad (10)$$

Note that this inequality is stronger than strictly needed (at least for nonlinear systems), but for pedagogical reasons we choose this simpler formulation. For the perturbed model, the inequality in (10) can in certain cases (including linear systems) be modified to

$$\frac{\partial V}{\partial x} f(x, h(x)) \leq -\|x\|^2 + \beta \|e\|^2 \qquad (11)$$

in which $\beta > 0$ is a constant used to indicate how the presence of the implementation error e affects the decrease of the Lyapunov function. Based on (10) one can now choose the function C in (5) to preserve GAS of the event-triggered implementation. For instance, $C(x(t), x(t_k)) = \|x(t_k) - x(t)\| - \sigma \|x(t)\|$, i.e.,

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| > \sigma \|x(t)\|\}, \qquad (12)$$

assures that

$$\|e\| \leq \sigma \|x\| \qquad (13)$$

holds. When $\sigma < 1/\beta$, we obtain from (11) and (13) that GAS properties are preserved for the event-triggered implementation. Besides, under certain conditions provided in Tabuada (2007), a global positive lower bound exists on the inter-execution times, i.e., there exists a $\tau_{\min} > 0$ such that $t_{k+1} - t_k > \tau_{\min}$ for all $k \in \mathbb{N}$ and all initial states x_0.

Also self-triggered controllers can be derived using the perturbation approach. In this case, stability properties can be guaranteed by choosing M in (6) ensuring that $C(x(t), x(t_k)) \leq 0$ holds for all times $t \in [t_k, t_{k+1})$ and all $k \in \mathbb{N}$.

Hybrid System Approach

By taking as a state variable $\xi = (x, e)$, one can write the closed-loop event-triggered control system given by (1), (3), and (5) as the hybrid impulsive system (Goebel et al. 2009)

$$\dot{\xi} = \begin{pmatrix} f(x, h(x + e)) \\ -f(x, h(x + e)) \end{pmatrix} \text{ when } C(x, x + e) \geq 0 \qquad (14a)$$

$$\xi^+ = \begin{pmatrix} x \\ 0 \end{pmatrix} \text{ when } C(x, x + e) \leq 0. \qquad (14b)$$

This observation was made in Donkers and Heemels (2010, 2012) and Postoyan et al. (2011). Tools from hybrid system theory can be used to analyze this model, which is more accurate as it includes the error dynamics of the event-triggered closed-loop system. In fact, the stability bounds obtained via the hybrid system approach can be proven to be never worse than ones obtained using the perturbation approach in many cases, see, e.g., Donkers and Heemels (2012), and typically the hybrid system approach provides (strictly) better results in practice. However, in general an analysis via the hybrid system approach is more complicated than using a perturbation approach.

Note that by including a time variable τ, one can also write the closed-loop system corresponding to self-triggered control (1), (3), and (6) as a hybrid system using the state variable $\chi = (x, e, \tau)$. This leads to the model

E

$$\dot{\chi} = \begin{pmatrix} f(x, h(x+e)) \\ -f(x, h(x+e)) \\ 1 \end{pmatrix} \text{when} 0 \leq \tau \leq M(x+e)$$

$$(15a)$$

$$\chi^+ = \begin{pmatrix} x \\ 0 \\ 0 \end{pmatrix} \text{ when } \tau = M(x+e), \qquad (15b)$$

which can be used for analysis based on hybrid tools as well.

Alternative Event-Triggering Mechanisms

There are various alternative event-triggering mechanisms. A few of them are described in this section.

Relative, Absolute, and Mixed Triggering Conditions

Above we discussed a very basic event-triggering condition in the form given in (12), which is sometimes called *relative* triggering as the next control task is executed at the instant when the ratio of the norms of the error $\|e\|$ and the measured state $\|x\|$ is larger than or equal to σ. Also *absolute* triggering of the form

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| \geq \delta\} \qquad (16)$$

can be considered. Here $\delta > 0$ is an absolute threshold, which has given this scheme the name send-on-delta (Miskowicz 2006). Recently, a *mixed* triggering mechanism of the form

$$t_{k+1} = \inf\{t > t_k \mid \|e(t)\| \geq \sigma\|x(t)\| + \delta\},$$

$$(17)$$

combining an absolute and a relative threshold, was proposed (Donkers and Heemels 2012). It is particularly effective in the context of output-based control.

Model-Based Triggering

In the triggering conditions discussed so far, essentially the current control value $u(t)$ is based on a *held* value $x(t_k)$ of the state variable, as specified in (3). However, if good model-based information regarding the plant is available, one can use better model-based predictions of the actuator signal. For instance, in the linear context, (Lunze and Lehmann 2010) proposed to use a *control input generator* instead of a plain zero-order hold function. In fact, the plant model was described by

$$\dot{x} = Ax + Bu + Ew \qquad (18)$$

with $x \in \mathbb{R}^{n_x}$ the state variable, $u \in \mathbb{R}^{n_u}$ the input variable, and $w \in \mathbb{R}^{n_w}$ a bounded disturbance input. It was assumed that a well functioning state feedback controller $u = Kx$ was available. The control input generator was then based on the model-based predictions given for $[t_k, t_{k+1})$ by

$$\dot{x}_s(t) = (A + BK)x_s(t) + E\hat{w}(t_k)$$

$$\text{with } x_s(t_k) = x(t_k) \qquad (19)$$

and $\hat{w}(t_k)$ is an estimate for the (average) disturbance value, which is determined at execution time t_k, $k \in \mathbb{N}$. The applied input to the actuator is then given by $u(t) = Kx_s(t)$ for $t \in [t_k, t_{k+1})$, $k \in \mathbb{N}$. Note that (19) provides a prediction of the closed-loop state evolution using the latest received value of the state $x(t_k)$ and the estimate $\hat{w}(t_k)$ of the disturbances. Also the event-triggering condition is based on this model-based prediction of the state as it is given by

$$t_{k+1} = \inf\{t > t_k \mid \|x_s(t) - x(t)\| \geq \delta\}. \quad (20)$$

Hence, when the prediction $x_s(t)$ diverts to far from the measured state $x(t)$, the next event is triggered so that updates of the state are sent to the actuator. These model-based triggering schemes can enhance the communication savings as they reduce the number of events by using model-based knowledge.

Other model-based event-triggered control schemes are proposed, for instance, in Yook et al. (2002), Garcia and Antsaklis (2013), and Heemels and Donkers (2013).

Triggering with Time-Regularization

Time-regularization was proposed for *output-based* triggering to avoid the occurrence of accumulations in the execution times (Zeno behavior) that would obstruct the existence of a positive lower bound on the inter-execution times $t_{k+1} - t_k$, $k \in \mathbb{N}$. In Tallapragada and Chopra (2012a,b), the triggering update

$$t_{k+1} = \inf\{t > t_k + T \mid \|e(t)\| \geq \sigma\|x(t)\|\} \tag{21}$$

was proposed, where $T > 0$ is a built-in lower bound on the minimal inter-execution times. The authors discussed how T and σ can be designed to guarantee closed-loop stability. In Heemels et al. (2008) a similar triggering was proposed using an absolute-type of triggering.

An alternative to exploiting a built-in lower bound T is combining ideas from time-triggered control and event-triggering control. Essentially, the idea is to only verify a specific event-triggering condition at certain equidistant time instants kT_s, $k \in \mathbb{N}$, where $T_s > 0$ is the sampling period. Such proposals were mentioned in, for instance, Årzén (1999), Yook et al. (2002), Henningsson et al. (2008), and Heemels et al. (2008, 2013). In this case the execution times are given by

$$t_{k+1} = \inf\{t > t_k \mid t = kT_s, \ k \in \mathbb{N},$$
$$\text{and } \|e(t)\| \geq \sigma\|x(t)\|\} \tag{22}$$

in case a relative triggering is used. In Heemels et al. (2013) the term *periodic event-triggered control* was coined for this type of control.

Decentralized Triggering Conditions

Another important extension of the mentioned event-triggered controllers, especially in large-scale networked systems, is the decentralization of the event-triggered control. Indeed, if one focuses on any of the abovementioned event-triggering conditions (take, e.g., (5)), it is obvious that the full state variable $x(t)$ has to be continuously available in a central coordinator to determine if an event is triggered or not. If the sensors that measure the state are physically distributed over a wide area, this assumption is prohibitive for its implementation. In such cases, it is of high practical importance that the event-triggering mechanism can be decentralized and the execution of control tasks can be executed based on local information. One first idea could be to use *local* event-triggering mechanisms for the i-th sensor that measures x_i. One could "decentralize" the condition (5), into

$$t_{k^i+1}^i = \inf\{t > t_{k^i}^i \mid \|e_i(t)\| \geq \sigma\|x_i(t)\|\}, \tag{23}$$

in which $e_i(t) = x_i(t_{k^i}^i) - x_i(t)$ for $t \in [t_{k^i}^i, t_{k^i+1}^i)$, $k^i \in \mathbb{N}$. Note that each sensor now has its own execution times $t_{k^i}^i$, $k^i \in \mathbb{N}$ at which the information $x_i(t)$ is transmitted. More importantly, the triggering condition (23) is based on local data only and does not need a central coordinator having access to the complete state information. Besides since (23) still guarantees that (13) holds, stability properties can still be guaranteed; see Mazo and Tabuada (2011).

Several other proposals for decentralized event-triggered control schemes were made, e.g., Persis et al. (2013), Wang and Lemmon (2011), Garcia and Antsaklis (2013), Yook et al. (2002), and Donkers and Heemels (2012).

Triggering for Multi-agent Systems

Event-triggered control strategies are suitable for cooperative control of multi-agent systems. In multi-agent systems, local control actions of individual agents should lead to a desirable global behavior of the overall system. A prototype problem for control of multi-agent systems is the agreement problem (also called the consensus or rendezvous problem), where the states of all agents should converge to a common value (sometimes the average of the agents' initial conditions). The agreement problem has been shown to be solvable for certain low-order dynamical agents in both continuous and discrete time, e.g., Olfati-Saber et al. (2007). It was recently shown in Dimarogonas et al. (2012), Shi and Johansson (2011), and Seyboth et al. (2013) that the agreement problem can be solved using event-triggered control. In Seyboth et al. (2013)

the triggering times for agent i are determined by

$$t^i_{k^i+1} = \inf\{t > t^i_{k^i} \mid C_i(x_i(t), x_i(t^i_{k^i})) > 0\}, \tag{24}$$

which should be compared to the triggering times as specified through (5). The triggering condition compares the current state value with the one previously communicated, similarly to the previously discussed decentralized event-triggered control (see (23)), but now the communication is *only* to the agent's neighbors. Using such event-triggered communication, the convergence rate to agreement (i.e., $\|x_i(t) - x_j(t)\| \rightarrow 0$ as $t \rightarrow \infty$ for all i, j) can be maintained with a much lower communication rate than for time-triggered communication.

Outlook

Many simulation and experimental results show that event-triggered and self-triggered control strategies are capable of reducing the number of control task executions, while retaining a satisfactory closed-loop performance. In spite of these results, the actual deployment of these novel control paradigms in relevant applications is still rather marginal. Some exceptions include recent event-triggered control applications in underwater vehicles (Teixeira et al. 2010), process control (Lehmann et al. 2012), and control over wireless networks (Araujo et al. 2014). To foster the further development of event-triggered and self-triggered controllers in the future, it is therefore important to validate these strategies in practice, next to building up a complete system theory for them. Regarding the latter, it is fair to say that, even though many interesting results are currently available, the system theory for event-triggered and self-triggered control is far from being mature, certainly compared to the vast literature on time-triggered (periodic) sampled-data control. As such, many theoretical and practical challenges are ahead of us in this appealing research field.

Cross-References

Bibliography

Araujo J, Mazo M Jr, Anta A, Tabuada P, Johansson KH (2014) System architectures, protocols, and algorithms for aperiodic wireless control systems. Industrial Informatics, IEEE Trans Ind Inform. 10(1): 175–184

Årzén K-E (1999) A simple event-based PID controller. In: Proceedings of the IFAC World Congress, Beijing, China, vol 18, pp 423–428. Preprints

Åström KJ, Bernhardsson BM (1999) Comparison of periodic and event based sampling for first order stochastic systems. In: Proceedings of the IFAC World Congress, Beijing, China, pp 301–Ű306

Astrőm, KJ, Wittenmark B (1997) Computer controlled systems. Prentice Hall, Upper Saddle River

De Persis C, Sailer R, Wirth F (2013) Parsimonious event-triggered distributed control: a Zeno free approach. Automatica 49(7):2116–2124

Dimarogonas DV, Frazzoli E, Johansson KH (2012) Distributed event-triggered control for multi-agent systems. IEEE Trans Autom Control 57(5):1291–1297

Donkers MCF, Heemels WPMH (2010) Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved event-triggering. In: Proceedings of the IEEE conference on decision and control, Atlanta, Georgia, USA, pp 3246–3251

Donkers MCF, Heemels WPMH (2012) Output-based event-triggered control with guaranteed $\mathcal{L}_\infty$-gain and improved and decentralised event-triggering. IEEE Trans Autom Control 57(6): 1362–1376

Franklin GF, Powel JD, Emami-Naeini A (2010) Feedback control of dynamical systems. Prentice Hall, Upper Saddle River

Garcia E, Antsaklis PJ (2013) Model-based event-triggered control for systems with quantization and time-varying network delays. IEEE Trans Autom Control 58(2):422–434

Goebel R, Sanfelice R, Teel AR (2009) Hybrid dynamical systems. IEEE Control Syst Mag 29: 28–93

Gupta S (1963) Increasing the sampling efficiency for a control system. IEEE Trans Autom Control 8(3): 263–264

Heemels WPMH, Donkers MCF (2013) Model-based periodic event-triggered control for linear systems. Automatica 49(3):698–711

Heemels WPMH, Sandee JH, van den Bosch PPJ (2008) Analysis of event-driven controllers for linear systems. Int J Control 81:571–590

Heemels WPMH, Donkers MCF, Teel AR (2013) Periodic event-triggered control for linear systems. IEEE Trans Autom Control 58(4):847–861

Henningsson T, Johannesson E, Cervin A (2008) Sporadic event-based control of first-order linear stochastic systems. Automatica 44:2890–2895

Lehmann D, Kiener GA, Johansson KH (2012) Event-triggered PI control: saturating actuators and anti-windup compensation. In: Proceedings of the IEEE conference on decision and control, Maui

Lunze J, Lehmann D (2010) A state-feedback approach to event-based control. Automatica 46:211–215

Mazo M Jr, Tabuada P (2011) Decentralized event-triggered control over wireless sensor/actuator networks. IEEE Trans Autom Control 56(10):2456–2461. Special issue on Wireless Sensor and Actuator Networks

Miskowicz M (2006) Send-on-delta concept: an event-based data-reporting strategy. Sensors 6: 49–63

Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

Postoyan R, Anta A, Nešić D, Tabuada P (2011) A unifying Lyapunov-based framework for the event-triggered control of nonlinear systems. In: Proceedings of the joint IEEE conference on decision and control and European control conference, Orlando, pp 2559–2564

Seyboth GS, Dimarogonas DV, Johansson KH (2013) Event-based broadcasting for multi-agent average consensus. Automatica 49(1):245–252

Shi G, Johansson KH (2011) Multi-agent robust consensus—part II: application to event-triggered coordination. In: Proceedings of the IEEE conference on decision and control, Orlando

Tabuada P (2007) Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans Autom Control 52(9):1680–1685

Tallapragada P, Chopra N (2012a) Event-triggered decentralized dynamic output feedback control for LTI systems. In: IFAC workshop on distributed estimation and control in networked systems, pp 31–36

Tallapragada P, Chopra N (2012b) Event-triggered dynamic output feedback control for LTI systems. In: IEEE 51st annual conference on decision and control (CDC), Maui, pp 6597–6602

Teixeira PV, Dimarogonas DV, Johansson KH, Borges de Sousa J (2010) Event-based motion coordination of multiple underwater vehicles under disturbances. In: IEEE OCEANS, Sydney

Velasco M, Marti P, Fuertes JM (2003) The self triggered task model for real-time control systems. In: Proceedings of 24th IEEE real-time systems symposium, work-in-progress session, Cancun, Mexico

Wang X, Lemmon MD (2011) Event-triggering in distributed networked systems with data dropouts and delays. IEEE Trans Autom Control 586–601

Yook JK, Tilbury DM, Soparkar NR (2002) Trading computation for bandwidth: reducing communication in distributed control systems using state estimators. IEEE Trans Control Syst Technol 10(4): 503–518

Evolutionary Games

Eitan Altman
INRIA, Sophia-Antipolis, France

Abstract

Evolutionary games constitute the most recent major mathematical tool for understanding, modelling and predicting evolution in biology and other fields. They complement other well establlished tools such as branching processes and the Lotka-Volterra (1910) equations (e.g. for the predator - prey dynamics or for epidemics evolution). Evolutionary Games also brings novel features to game theory. First, it focuses on the dynamics of competition rather than restricting attention to the equilibrium. In particular, it tries to explain how an equilibrium emerges. Second, it brings new definitions of stability, that are more adapted to the context of large populations. Finally, in contrast to standard game theory, players are not assumed to be "rational" or "knowledgeable" as to anticipate the other players' choices. The objective of this article, is to present foundations as well as recent advances in evolutionary games, highlight the novel concepts that they introduce with respect

to game theory as formulated by John Nash, and describe through several examples their huge potential as tools for modeling interactions in complex systems.

Keywords

Evolutionary stable strategies; Fitness; Replicator dynamics

Introduction

Evolutionary game theory is the youngest of several mathematical tools used in describing and modeling evolution. It was preceded by the theory of branching processes (Watson and Francis Galton 1875) and its extensions (Altman 2008) which have been introduced in order to explain the evolution of family names in the English population of the second half of the nineteenth century. This theory makes use of the probabilistic distribution of the number of offspring of an individual in order to predict the probability at which the whole population would become eventually extinct. It describes the evolution of the number of offsprings of a given individual. The Lotka-Volterra equations (Lotka-Volterra 1910) and their extensions are differential equations that describe the population size of each of several species that have a predator-prey type relation. One of the foundations in evolutionary games (and its extension to population games) which is often used as the starting point in their definition is the replicator dynamics which, similarly to the Lotka-Volterra equations, describe the evolution of the size of various species that interact with each other (or of various behaviors within a given population). In both the Lotka-Volterra equations and in replicator dynamics, the evolution of the size of one type of population may depend on the sizes of all other populations. Yet, unlike the Lotka-Volterra equations, the object of the modeling is the normalized sizes of populations rather than the size itself. By normalized size of some type, we mean the fraction of that type within the whole population. A basic feature in

evolutionary games is, thus, that the evolution of the fraction of a given type in the population depends on the sizes of other types only through the normalized size rather than through their actual one.

The relative rate of the decrease or increase of the normalized population size of some type in the replicator dynamics is what we call fitness and is to be understood in the Darwinian sense. If some type or some behavior increases more than another one, then it has a larger fitness. the evolution of the fitness as described by the replicator dynamics is a central object of study in evolutionary games.

So far we did not actually consider any game and just discussed ways of modeling evolution. The relation to game theory is due to the fact that under some conditions, the fitness converges to some fixed limit, which can be identified as an equilibrium of a matrix game in which the utilities of the players are the fitnesses. This limit is then called an ESS – evolutionary stable strategy – as defined by Meynard Smith and Price in Maynard Smith and Price (1973). It can be computed using elementary tools in matrix games and then used for predicting the (long term) distribution of behaviors within a population. Note that an equilibrium in a matrix game can be obtained only when the players of the matrix game are rational (each one maximizing its expected utility, being aware of the utilities of other players and of the fact that these players maximize their utilities, etc.). A central contribution of evolutionary games is thus to show that evolution of possibly nonrational populations converges under some conditions to the equilibrium of a game played by rational players. This surprising relationship between the equilibrium of a noncooperative matrix game and the limit points of the fitness dynamics has been supported by a rich body of experimental results; see Friedman (1996).

On the importance of the ESS for understanding the evolution of species, Dawkins writes in his book "The Selfish Gene" (Dawkins 1976): "we may come to look back on the invention of the ESS concept as one of the most important

advances in evolutionary theory since Darwin." He further specifies: "Maynard Smith's concept of the ESS will enable us, for the first time, to see clearly how a collection of independent selfish entities can come to resemble a single organized whole."

Here we shall follow the nontraditional approach describing evolutionary games: we shall first introduce the replicator dynamics and then introduce the game theoretic interpretation related to it.

Replicator Dynamics

In the biological context, the replicator dynamics is a differential equation that describes the way in which the usage of strategies changes in time. They are based on the idea that the average growth rate per individual that uses a given strategy is proportional to the excess of fitness of that strategy with respect to the average fitness.

In engineering, the replicator dynamics could be viewed as a rule for updating mixed strategies by individuals. It is a decentralized rule since it only requires knowing the average utility of the population rather than the strategy of each individual.

Replicator dynamics is one of the most studied dynamics in evolutionary game theory. It has been introduced by Taylor and Jonker (1978). The replicator dynamics has been used for describing the evolution of road traffic congestion in which the fitness is determined by the strategies chosen by all drivers (Sandholm 2009). It has also been studied in the context of the association problem in wireless communications (Shakkottai et al. 2007).

Consider a set of N strategies and let $p_j(t)$ be the fraction of the whole population that uses strategy j at time t. Let $p(t)$ be the corresponding N-dimensional vector. A function f_j is associated with the growth rate of strategy j, and it is assumed to depend on the fraction of each of the N strategies in the population. There are various forms of replicator dynamics (Sandholm 2009) and we describe here the one most commonly used. It is given by

$$\dot{p}_j(t) = \mu\, p_j(t) \left[f_j(p(t)) - \sum_{k=1}^{N} p_k(t) f_k(p(t)) \right],$$
(1)

where μ is some positive constant and the payoff function f_k is called the fitness of strategy k.

In evolutionary games, evolution is assumed to be due to pairwise interactions between players, as will be described in the next section. Therefore, f_k has the form $f_k(p) = \sum_{i=1}^{N} J(k,i) p(i)$ where $J(k,i)$ is the fitness of an individual playing k if it interacts with an individual that plays strategy i.

Within quite general settings (Weibull 1995), the above replicator dynamics is known to converge to an ESS (which we introduce in the next section).

Evolutionary Games: Fitnesses

Consider an infinite population of players. Each individual i plays at times t_n^i, $n = 1, 2, 3, \ldots$ (assumed to constitute an independent Poisson process with some rate λ) a matrix game against some player $j(n)$ randomly selected within the population. The choice $j(n)$ of the other players at different times is independent. All players have the same finite space of pure strategies (also called actions) K. Each time it plays, a player may use a mixed strategy p, i.e., a probability measure over the set of pure strategies. We consider $J(k,i)$ (defined in the previous section) to be the payoff for a tagged individual if it uses a strategy k, and it interacts with an individual using strategy i. With some abuse of notation, one denotes by $J(p,q)$ the expected payoff for a player who uses a mixed strategy p when meeting another individual who adopts the mixed strategy q. If we define a payoff matrix A and consider p and q to be column vectors, then $J(p,q) = p'Aq$. The payoff function J is indeed linear in p and q. A strategy q is called a Nash equilibrium if

$$\forall p \in \Delta(K), \quad J(q,q) \geq J(p,q) \qquad (2)$$

where $\Delta(K)$ is the set of probabilities over the set K.

Suppose that the whole population uses a strategy q and that a small fraction ϵ (called "mutations") adopts another strategy p. Evolutionary forces are expected to select against p if

$$J(q, \epsilon p + (1 - \epsilon)q) > J(p, \epsilon p + (1 - \epsilon)q). \quad (3)$$

Evolutionary Stable Strategies: ESS

Definition 1 q is said to be an evolutionary stable strategy (ESS) if for every $p \neq q$ there exists some $\overline{\epsilon}_p > 0$ such that (3) holds for all $\epsilon \in (0, \overline{\epsilon}_p)$.

The definition of ESS is thus related to a robustness property against deviations by a whole (possibly small) fraction of the population. This is an important difference that distinguishes the equilibrium in populations as seen by biologists and the standard Nash equilibrium often used in economics context, in which robustness is defined against the possible deviation of a single user. Why do we need the stronger type of robustness? Since we deal with large populations, it is likely to be expected that from time to time, some group of individuals may deviate. Thus robustness against deviations by a single user is not sufficient to ensure that deviations will not develop and end up being used by a growing portion of the population.

Often ESS is defined through the following equivalent definition.

Theorem 1 (**Weibull 1995, Proposition 2.1 or Hofbauer and Sigmund 1998, Theorem 6.4.1, p 63**) *A strategy q is said to be an evolutionary stable strategy if and only if $\forall p \neq q$ one of the following conditions holds:*

$$J(q, q) > J(p, q), \quad (4)$$

or

$$J(q, q) = J(p, q) \text{ and } J(q, p) > J(p, p). \quad (5)$$

In fact, if condition (4) is satisfied, then the fraction of mutations in the population will tend to decrease (as it has a lower fitness, meaning a lower growth rate). Thus, the strategy q is then immune to mutations. If it does not but if still the condition (5) holds, then a population using q is "weakly" immune against a mutation using p. Indeed, if the mutant's population grows, then we shall frequently have individuals with strategy q competing with mutants. In such cases, the condition $J(q, p) > J(p, p)$ ensures that the growth rate of the original population exceeds that of the mutants.

A mixed strategy q that satisfies (4) for all $p \neq q$ is called strict Nash equilibrium. Recall that a mixed strategy q that satisfies (2) for all $p \neq q$ is a Nash equilibrium. We conclude from the above theorem that being a strict Nash equilibrium implies being an ESS, and being an ESS implies being a Nash equilibrium. Note that whereas a mixed Nash equilibrium is known to exist in a matrix game, an ESS may not exist. However, an ESS is known to exist in evolutionary games where the number of strategies available to each player is 2 (Weibull 1995).

Proposition 1 *In a symmetric game with two strategies for each player and no pure Nash equilibrium, there exists a unique mixed Nash equilibrium which is an ESS.*

Example: The Hawk and Dove Game

We briefly describe the hawk and dove game (Maynard Smith and Price 1973). A bird that searches food finds itself competing with another bird over food and has to decide whether to adopt a peaceful behavior (dove) or an aggressive one (hawk). The advantage of behaving aggressively is that in an interaction with a peaceful bird, the aggressive one gets access to all the food. This advantage comes at a cost: a hawk which meets another hawk ends up fighting with it and thus takes a risk of getting wounded. In contrast, two doves that meet in a contest over food share it without fighting. The fitnesses for player 1 (who chooses a row) are summarized in Table 1, in which the cost for fighting is taken to be some parameter $\delta > 1/2$.

This game has a unique mixed Nash equilibrium (and thus a unique ESS) in which the fraction p of aggressive birds is given by

Evolutionary Games, Table 1 The hawk-dove game

	H	D
H	$1/2 - \delta$	1
D	0	1/2

$$p = \frac{2}{1.5 + \delta}$$

Extension: Evolutionary Stable Sets

Assume that there are two mixed strategies p_i and p_j that have the same performance against each other, i.e., $J(p_i, p_j) = J(p_j, p_j)$. Then neither one of them can be an ESS, even if they are quite robust against other strategies. Now assume that when excluding one of them from the set of mixed strategies, the other one is an ESS. This could imply that different combinations of these two ESS's could coexist and would together be robust to any other mutations. This motivates the following definition of an ESSet (Cressman 2003):

Definition 2 A set E of symmetric Nash equilibria is an evolutionarily stable set (ESSet) if, for all $q \in E$, we have $J(q, p) > J(p, p)$ for all $p \notin E$ and such that $J(p, q) = J(q, q)$.

Properties of ESSet:

 (i) For all p and p' in an ESSet E, we have $J(p', p) = J(p, p)$.
 (ii) If a mixed strategy is an ESS, then the singleton containing that mixed strategy is an ESSet.
(iii) If the ESSet is not a singleton, then there is no ESS.
(iv) If a mixed strategy is in an ESSet, then it is a Nash equilibrium (see Weibull 1995, p. 48, Example 2.7).
 (v) Every ESSet is a disjoint union of Nash equilibria.
(vi) A perturbation of a mixed strategy which is in the ESSet can move the system to another mixed strategy in the ESSet. In particular, every ESSet is asymptotically stable for the replicator dynamics (Cressman 2003).

Summary and Future Directions

The entry has provided an overview of the foundations of evolutionary games which include the ESS (evolutionary stable strategy) equilibrium concept that is stronger than the standard Nash equilibrium and the modeling of the dynamics of the competition through the replicator dynamics. Evolutionary game framework is a first step in linking game theory to evolutionary processes. The payoff of a player is identified as its fitness, i.e., the rate of reproduction. Further development of this mathematical tool is needed for handling hierarchical fitness, i.e., the cases where the individual that interacts cannot be directly identified with the reproduction as it is part of a larger body. For example, the behavior of a blood cell in the human body when interacting with a virus cannot be modeled as directly related to the fitness of the blood cell but rather to that of the human body. A further development of the theory of evolutionary games is needed to define meaningful equilibrium notions and relate them to replication in such contexts.

Cross-References

▶ Dynamic Noncooperative Games
▶ Game Theory: Historical Overview

Recommended Reading

Several books cover evolutionary game theory well. These include Cressman (2003), Hofbauer and Sigmund (1998), Sandholm (2009), Vincent and Brown (2005), and Weibull (1995). In addition, the book *The Selfish Gene* by Dawkins presents an excellent background on evolution in biology.

Bibliography

Altman E (2008) Semi-linear stochastic difference equations. Discret Event Dyn Syst 19:115–136

Cressman R (2003) Evolutionary dynamics and extensive form games. MIT, Cambridge

Dawkins R (1976) The selfish gene. Oxford University Press, Oxford

Friedman D (1996) Equilibrium in evolutionary games: some experimental results. Econ J 106: 1–25

Hofbauer J, Sigmund K (1998) Evolutionary games and population dynamics. Cambridge University Press, Cambridge/New York

Lotka-Volterra AJ (1910) Contribution to the theory of periodic reaction. J Phys Chem 14(3): 271–274

Maynard Smith J, Price GR (1973) The logic of animal conflict. Nature 246(5427):15–18

Sandholm WH (2009) Population games and evolutionary dynamics. MIT

Shakkottai S, Altman E, Kumar A (2007) Multihoming of users to access points in WLANs: a population game perspective. IEEE J Sel Areas Commun Spec Issue Non-Coop Behav Netw 25(6):1207–1215

Taylor P, Jonker L (1978) Evolutionary stable strategies and game dynamics. Math Biosci 16: 76–83

Vincent TL, Brown JS (2005) Evolutionary game theory, natural selection & Darwinian dynamics. Cambridge University Press, Cambridge

Watson HW, Galton F (1875) On the probability of the extinction of families. J Anthropol Inst Great Br 4: 138–144

Weibull JW (1995) Evolutionary game theory. MIT, Cambridge

Experiment Design and Identification for Control

Håkan Hjalmarsson
School of Electrical Engineering, ACCESS
Linnaeus Center, KTH Royal Institute of
Technology, Stockholm, Sweden

Abstract

Understanding the effect of experiment on estimation result is a crucial part of system identification – if the experiment is constrained or otherwise fixed, then the implied limitations need to be understood – but if the experiment can be designed, then given its fundamental importance that design parameter should be fully exploited, this entry will give an understanding of how it can be exploited. We also briefly discuss the particulars of identification for model-based control, one of the main applications of system identification.

Keywords

Introduction

The accuracy of an identified model is governed by:

(i) Information content in the data used for estimation

(ii) The complexity of the model structure

The former is related to the noise properties and the "energy" of the external excitation of the system and how it is distributed. In regard to (ii), a model structure which is not flexible enough to capture the true system dynamics will give rise to a systematic error, while an overly flexible model will be overly sensitive to noise (so-called overfitting). The model complexity is closely associated with the number of parameters used. For a linear model structure with n parameters modeling the dynamics, it follows from the invariance result in Rojas et al. (2009) that to obtain a model for which the variance of the frequency function estimate is less than $1/\gamma$ over all frequencies, the signal-to-noise ratio, as measured by input energy over noise variance, must be at least $n\gamma$. With energy being power $\times$ time and as input power is limited in physical systems, this indicates that the experiment time grows at least linearly with the number of model parameters. When the input energy budget is limited, the only way around this problem is to sacrifice accuracy over certain frequency intervals. The methodology to achieve this in a systematic way is known as experiment design.

Model Quality Measures

The Cramér-Rao bound provides a lower bound on the covariance matrix of the estimation error for an unbiased estimator. With $\hat{\theta}_N \in \mathbb{R}^n$ denoting the parameter estimate (based on N input–output samples) and θ_o the true parameters,

$$N\mathrm{E}\left[\left(\hat{\theta}_N - \theta_o\right)\left(\hat{\theta}_N - \theta_o\right)^T\right] \geq N\ I_F^{-1}(\theta_o, N) \tag{1}$$

where $I_F(\theta_o, N) \in \mathbb{R}^{n \times n}$ appearing in the lower bound is the so-called Fisher information matrix (Ljung 1999). For consistent estimators, i.e., when $\hat{\theta}_N \rightarrow \theta_o$ as $N \rightarrow \infty$, the inequality (1) typically holds asymptotically as the sample size N grows to infinity. The right-hand side in (1) is then replaced by the inverse of the per sample Fisher information $I_F(\theta_o) := \lim_{N\rightarrow\infty} I_F(\theta_o, N)/N$. An estimator is said to be asymptotically efficient if equality is reached in (1) as $N \rightarrow \infty$.

Even though it is possible to reduce the mean-square error by constraining the model flexibility appropriately, it is customary to use consistent estimators since the theory for biased estimators is still not well understood. For such estimators, using some function of the Fisher information as performance measure is natural.

General-Purpose Quality Measures

Over the years a number of "general-purpose" quality measures have been proposed. Perhaps the most frequently used is the determinant of the inverse Fisher information. This represents the volume of confidence ellipsoids for the parameter estimates and minimizing this measure is known as D-optimal design. Two other criteria relating to confidence ellipsoids are E-optimal design, which uses the length of the longest principal axis (the minimum eigenvalue of I_F) as quality measure, and A-optimal design, which uses the sum of the squared lengths of the principal axes (the trace of I_F^{-1}).

Application-Oriented Quality Measures

When demands are high and/or experimentation resources are limited, it is necessary to tailor the experiment carefully according to the intended use of the model. Below we will discuss a couple of closely related application-oriented measures.

Average Performance Degradation

Let $V_{\mathrm{app}}(\theta) \geq 0$ be a measure of how well the model corresponding to parameter θ performs when used in the application. In finance, V_{app} can, e.g., represent the ability to predict the stock market. In process industry, V_{app} can represent the profit gained using a feedback controller based on the model corresponding to θ. Let us assume that V_{app} is normalized such that $\min_\theta V_{\mathrm{app}}(\theta) = V_{\mathrm{app}}(\theta_o) = 0$. That V_{app} has minimum corresponding to the parameters of the true system is quite natural. We will call V_{app} the application cost. Assuming that the estimator is asymptotically efficient, using a second-order Taylor approximation gives that the average application cost can be expressed as (the first-order term vanishes since θ_o is the minimizer of V_{app})

$$\mathrm{E}\left[V_{\mathrm{app}}(\hat{\theta}_N)\right] \approx \frac{1}{2}\mathrm{E}\left[\left(\hat{\theta}_N - \theta_o\right)^T V_{\mathrm{app}}''(\theta_o)\left(\hat{\theta}_N - \theta_o\right)\right]$$
$$= \frac{1}{2N}\mathrm{Tr}\left\{V_{\mathrm{app}}''(\theta_o)I_F^{-1}(\theta_o)\right\} \tag{2}$$

This is a generalization of the A-optimal design measure and its minimization is known as L-optimal design.

Acceptable Performance

Alternatively, one may define a set of acceptable models, i.e., a set of models which will give acceptable performance when used in the application. With a performance degradation measure defined of the type V_{app} above, this would be a level set

$$\mathcal{E}_{\mathrm{app}} = \left\{\theta :\ V_{\mathrm{app}}(\theta) \leq \frac{1}{\gamma}\right\} \tag{3}$$

for some constant $\gamma > 0$. The objective of the experiment design is then to ensure that the

resulting estimate ends up in $\mathcal{E}_{\mathrm{app}}$ with high probability.

Design Variables

In an identification experiment there are a number of design variables at the user's disposal. Below we discuss three of the most important ones.

Sampling Interval
For the sampling interval, the general advice from an information theoretic point of view is to sample as fast as possible (Ljung 1999). However, sampling much faster than the time constants of the system may lead to numerical issues when estimating discrete time models as there will be poles close to the unit circle. Downsampling may thus be required.

Feedback
Generally speaking, feedback has three effects from an identification and experiment design point of view:

 (i) Not all the power in the input can be used to estimate the system dynamics when a noise model is estimated as a part of the input signal has to be used for the latter task; see Section 8.1 in Forssell and Ljung (1999). When a very flexible noise model is used, the estimate of the system dynamics then has to rely almost entirely on external excitation.
 (ii) Feedback can reduce the effect of disturbances and noise at the output. When there are constraints on the outputs, this allows for larger (input) excitation and therefore more informative experiments.
(iii) The cross-correlation between input and noise/disturbances requires good noise models to avoid biased estimates (Ljung 1999).

Strictly speaking, (i) is only valid when the system and noise models are parametrized separately. Items (i) and (ii) imply that when there are constraints on the input only, then the optimal design is always in open loop, whereas for output constrained only problems, the experiment

should be conducted in closed loop (Agüero and Goodwin 2007).

External Excitation Signals
The most important design variable is the external excitation, including the length of the experiment. Even for moderate experiment lengths, solving optimal experiment design problems with respect to the entire excitation sequence can be a formidable task. Fortunately, for experiments of reasonable length, the design can be split up in two steps:

 (i) First, optimization of the probability density function of the excitation
 (ii) Generation of the actual sequence from the obtained density function through a stochastic simulation procedure

More details are provided in section "Computational Issues."

Experimental Constraints

An experiment is always subject to constraints, physical as well as economical. Such constraints are typically translated into constraints on the following signal properties:

 (i) *Variability.* For example, too high level of excitation may cause the end product to go off-spec, resulting in product waste and associated high costs.
 (ii) *Frequency content.* Often, too harsh movements of the inputs may damage equipment.
(iii) *Amplitudes.* For example, actuators have limited range, restricting input amplitudes.
(iv) *Waveforms.* In process industry, it is not uncommon that control equipment limit the type of signals that can be applied. In other applications, it may be physically possible to realize only certain types of excitation. See section "Waveform Generation" for further discussion.

It is also often desired to limit the experiment time so that the process may go back to normal operation, reducing, e.g., cost of personnel. The latter is especially important in the process industry where dynamics are slow. The above type of constraints can be formulated as constraints on

the design variables in section "Design Variables" and associated variables.

Experiment Design Criteria

There are two principal ways to define an optimal experiment design problem:

(i) *Best effort.* Here the best quality as, e.g., given by one of the quality measures in section "Model Quality Measures" is sought under constraints on the experimental effort and cost. This is the classical problem formulation.

(ii) *Least-costly.* The cheapest experiment is sought that results in a predefined model quality. Thus, as compared to best effort design, the optimization criterion and constraint are interchanged. This type of design was introduced by Bombois and coworkers; see Bombois et al. (2006).

As shown in Rojas et al. (2008), the two approaches typically lead to designs only differing by a scaling factor.

Computational Issues

The optimal experiment design problem based on the Fisher information is typically non-convex. For example, consider a finite-impulse response model subject to an experiment of length N with the measured outputs collected in the vector

$$Y = \Phi\theta + E, \Phi = \begin{bmatrix} u(0) & \dots & u(-(n-1)) \\ \vdots & \vdots & \vdots \\ u(N-1) & \dots & u(N-n) \end{bmatrix}$$

where $E \in \mathbb{R}^N$ is zero-mean Gaussian noise with covariance matrix $\sigma^2 I_{N \times N}$. Then it holds that

$$I_F(\theta_o, N) = \frac{1}{\sigma^2}\Phi^T\Phi \qquad (4)$$

From an experiment design point of view, the input vector $u = \begin{bmatrix} u(-(n-1)) \dots u(N) \end{bmatrix}^T$ is the design Variable, but with the elements of $I_F(\theta_o, N)$ being a quadratic function of the input

sequence, all typical quality measures become non-convex.

While various methods for non-convex numerical optimization can be used to solve such problems, they often encounter problems with, e.g., local minima. To address this a number of techniques have been developed where either the problem is reparametrized so that it becomes convex or where a convex approximation is used. The latter technique is called convex relaxation and is often based on a reparametrization as well. We use the example above to provide a flavor of the different techniques.

Reparametrization

If the input is constrained to be periodic so that $u(t) = u(t + N)$, $t = -n, \dots, -1$, it follows that the Fisher information is linear in the sample correlations of the input. Using these as design variables instead of u results in that all quality measures referred to above become convex functions.

This reparametrization thus results in the two-step procedure discussed in section "External Excitation Signals": First, the sample correlations are obtained from an optimal experiment design problem, and then an input sequence is generated that has this sample correlation. In the second step there is a considerable freedom. Notice, however, that since correlations do not directly relate to the actual amplitudes of the resulting signals, it is difficult to incorporate waveform constraints in this approach. On the contrary, variance constraints are easy to incorporate.

Convex Relaxations

There are several approaches to obtain convex relaxations.

Using the per Sample Fisher Information

If the input is a realization of a stationary random process and the sample size N is large enough, $I_F(\theta_o, N)/N$ is approximately equal to the per sample Fisher matrix which only depends on the correlation sequence of the input. Using this approximation, one can now follow the same procedure as in the reparametrization approach and first optimize the input correlation sequence.

The generation of a stationary signal with a certain correlation is a stochastic realization problem which can be solved using spectral factorization followed by filtering white noise sequence, i.e., a sequence of independent identically distributed random variables, through the (stable) spectral factor (Jansson and Hjalmarsson 2005).

More generally, it turns out that the per sample Fisher information for linear models/systems only depends on the joint input/noise spectrum (or the corresponding correlation sequence). A linear parametrization of this quantity thus typically leads to a convex problem (Jansson and Hjalmarsson 2005).

The set of all spectra is infinite dimensional and this precludes a search over all possible spectra. However, since there is a finite-dimensional parametrization of the per sample Fisher information (it is a symmetric $n \times n$ matrix), it is also possible to find finite-dimensional sets of spectra that parametrize all possible per sample Fisher information matrices. Multisines with appropriately chosen frequencies is one possibility. However, even though all per sample Fisher information matrices can be generated, the solution may be suboptimal depending on which constraints the problem contains.

The situation for nonlinear problems is conceptually the same, but here the entire probability density function of the stationary process generating the input plays the same role as the spectrum in the linear case. This is a much more complicated object to parametrize.

Lifting

An approach that can deal with amplitude constraints is based on a so-called lifting technique: Introduce the matrix $U = uu^T$, representing all possible products of the elements of u. This constraint is equivalent to

$$\begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} \geq 0, \quad \text{rank} \begin{bmatrix} U & u \\ u^T & 1 \end{bmatrix} = 1 \qquad (5)$$

The idea of lifting is now to observe that the Fisher information matrix is linear in the elements of U and by dropping the rank constraint in (5) a convex relaxation is obtained, where both

U and u (subject to the matrix inequality in (5)) are decision variables.

Frequency-by-Frequency Design

An approximation for linear systems that allows frequency-by-frequency design of the input spectrum and feedback is obtained by assuming that the model is of high order. Then the variance of an nth-order estimate, $G(e^{i\omega}, \hat{\theta}_N)$, of the frequency function can approximately be expressed as

$$\text{Var } G(e^{i\omega}, \hat{\theta}_N) \approx \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \qquad (6)$$

(▶ System Identification: An Overview) in the open loop case (there is a closed-loop extension as well), where Φ_u and Φ_v are the input and noise spectra, respectively. Performance measures of the type (2) can then be written as

$$\int_{-\pi}^{\pi} W(e^{i\omega}) \frac{\Phi_v(\omega)}{\Phi_u(\omega)} d\omega$$

where the weighting $W(e^{i\omega}) \geq 0$ depends on the application. When only variance constraints are present, such problems can be solved frequency by frequency, providing both simple calculations and insight into the design.

Implementation

We have used the notation $I_F(\theta_o, N)$ to indicate that the Fisher information typically (but not always) depends on the parameter corresponding to the true system. That the optimal design depends on the to-be identified system is a fundamental problem in optimal experiment design. There are two basic approaches to address this problem which are covered below. Another important aspect is the choice of waveform for the external excitation signal. This is covered last in this section.

Robust Experiment Design

In robust experiment design, it is assumed that it is known beforehand that the true parameter belongs to some set, i.e., $\theta_o \in \Theta$. A minimax

approach is then typically taken, finding the experiment that minimizes the worst performance over the set Θ. Such optimization problems are computationally very difficult.

Adaptive Experiment Design

The alternative to robust experiment design is to perform the design adaptively or sequentially, meaning that first a design is performed based on some initial "guess" of the true parameter, and then as samples are collected, the design is revised taking advantage of the data information. Interestingly, the convergence rate of the parameter estimate is typically sufficiently fast that for this approach the asymptotic distribution is the same as for the design based on the true model parameter (Hjalmarsson 2009).

Waveform Generation

We have argued above that it is the spectrum of the excitation (together with the feedback) that determines the achieved model accuracy in the linear time-invariant case. In section "Using the per Sample Fisher Information" we argued that a signal with a particular spectrum can be obtained by filtering a white noise sequence through a stable spectral factor of the desired spectrum. However, we have also in section "Experimental Constraints" argued that particular applications may require particular waveforms. We will here elaborate further on how to generate a waveform with desired characteristics.

From an accuracy point of view, there are two general issues that should be taken into account when the waveform is selected:

- *Persistence of excitation.* A signal with a spectrum having n nonzero frequencies (on the interval $(-\pi, \pi]$) can be used to estimate at most n parameters. Thus, as is typically the case, if there is uncertainty regarding which model structure to use before the experiment, one has to ensure that a sufficient number of frequencies is excited.
- *The crest factor.* For all systems, the maximum input amplitude, say A, is constrained. To deal with this from an experiment design point of view, it is convenient to introduce what is called the crest factor of a signal:

$$C_r^2 = \frac{\max_t u^2(t)}{\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} u^2(t)}$$

The crest factor is thus the ratio between the squared maximum amplitude and the power of the signal. Now, for a class of signal waveforms with a given crest factor, the input power that can be used is upper-bounded by

$$\lim_{N \to \infty} \frac{1}{N} \sum_{t=1}^{N} u^2(t) \leq \frac{A^2}{C_r^2} \tag{7}$$

However, the power is the integral of the signal spectrum, and since increasing the amplitude of the input signal spectrum will increase a model's accuracy, cf. (6), it is desirable to use as much signal power as possible. By (7) we see that this means that waveforms with low crest factor should be used.

A lower bound for the crest factor is readily seen to be 1. This bound is achieved for binary symmetric signals. Unfortunately, there exists no systematic way to design a binary sequence that has a prescribed spectrum. However, the so-called arcsin law may be used. It states that the sign of a zero-mean Gaussian process with correlation sequence r_τ gives a binary signal having correlation sequence $\tilde{r}_\tau = 2/\pi \arcsin(r_\tau)$. With $\tilde{r}_\tau$ given, one can try to solve this relation for the corresponding r_τ.

A crude, but often sufficient, method to generate binary sequences with desired spectral content is based on the use of *pseudorandom binary signals (PRBS)*. Such signals (which are generated by a shift register) are periodic signals which have correlation sequences similar to random white noise, i.e., a flat spectrum. By resampling such sequences, the spectrum can be modified. It should be noted that binary sequences are less attractive when it comes to identifying nonlinearities. This is easy to understand by considering a static system. If only one amplitude of the input is used, it will be impossible to determine whether the system is nonlinear or not.

A PRBS is a periodic signal and can therefore be split into its Fourier terms. With a period of M, each such term corresponds to one frequency on the grid $2\pi k/M$, $k = 0, \ldots, M - 1$. Such a signal can thus be used to estimate at most M parameters. Another way to generate a signal with period M is to add sinusoids corresponding to the above frequencies, with desired amplitudes. A periodic signal generated in this way is commonly referred to as a *MultiSine*. The crest factor of a multisine depends heavily on the relation between the phases of the sinusoids. times the number of sinusoids. It is possible to optimize the crest factor with respect to the choice of phases (Rivera et al. 2009). There exist also simple deterministic methods for choosing phases that give a good crest factor, e.g., Schroeder phasing. Alternatively, phases can be drawn randomly and independently, giving what is known as random-phase multisines (Pintelon and Schoukens 2012), a family of random signals with properties similar to Gaussian signals. Periodic signals have some useful features:

- *Estimation of nonlinearities.* A linear time-invariant system responds to a periodic input signal with a signal consisting of the same frequencies, but with different amplitudes and phases. Thus, it can be concluded that the system is nonlinear if the output contains other frequencies than the input. This can be explored in a systematic way to estimate also the nonlinear part of a system.

- *Estimation of noise variance.* For a linear time-invariant system, the difference in the output between different periods is due entirely to the noise if the system is in steady state. This can be used to devise simple methods to estimate the noise level.

- *Data compression.* By averaging measurements over different periods, the noise level can be reduced at the same time as the number of measurements is reduced.

Further details on waveform generation and general-purpose signals useful in system identification can be found in Pintelon and Schoukens (2012) and Ljung (1999).

Implications for the Identification Problem Per Se

In order to get some understanding of how optimal experimental conditions influence the identification problem, let us return to the finite-impulse response model example in section "Computational Issues." Consider a least-costly setting with an acceptable performance constraint. More specifically, we would like to use the minimum input energy that ensures that the parameter estimate ends up in a set of the type (3). An approximate solution to this is that a 99 % confidence ellipsoid for the resulting estimate is contained in $\mathcal{E}_{\text{app}}$. Now, it can be shown that a confidence ellipsoid is a level set for the average least-squares cost $\text{E}[V_N(\theta)] = \text{E}[\|Y - \Phi\theta\|^2] = \|\theta - \theta_o\|^2_{\Phi^T\Phi} + \sigma^2$. Assuming the application cost V_{app} also is quadratic in θ, it follows after a little bit of algebra (see Hjalmarsson 2009) that it must hold that

$$\text{E}[V_N(\theta)] \geq \sigma^2 \left(1 + \gamma c V_{\text{app}}(\theta)\right), \ \forall \theta \qquad (8)$$

for a constant c that is not important for our discussion. The value of $\text{E}[V_N(\theta)] = \|\theta - \theta_o\|^2_{\Phi^T\Phi} + \sigma^2$ is determined by how large the weighting $\Phi^T\Phi$ is, which in turn depends on how large the input u is. In a least-costly setting with the energy $\|u\|^2$ as criterion, the best solution would be that we have equality in (8). Thus we see that optimal experiment design tries to shape the identification criterion after the application cost. We have the following implications of this result:

(i) *Perform identification under appropriate scaling of the desired operating conditions.* Suppose that $V_{\text{app}}(\theta)$ is a function of how the system outputs deviate from a desired trajectory (determined by θ_o). Performing an experiment which performs the desired trajectory then gives that the sum of the squared prediction errors are an approximation of $V_{\text{app}}(\theta)$, at least for parameters close to θ_o. Obtaining equality in (8) typically requires an additional scaling

of the input excitation or the length of the experiment. The result is intuitively appealing: The desired operating conditions should reveal the system properties that are important in the application.

(ii) *Identification cost for application performance.* We see that the required energy grows (almost) linearly with γ, which is a measure of how close to the ideal performance (using the true parameter θ_o) we want to come. Furthermore, it is typical that as the performance requirements in the application increase, the sensitivity to model errors increases. This means that $V_{app}(\theta)$ increases, which thus in turn means that the identification cost increases. In summary, the identification cost will be higher, the higher performance that is required in the application. The inequality (8) can be used to quantify this relationship.

(iii) *Model structure sensitivity.* As V_{app} will be sensitive to system properties important for the application, while insensitive to system properties of little significance, with the identification criterion V_N matched to V_{app}, it is only necessary that the model structure is able to model the important properties of the system.

In any case, whatever model structure that is used, the identified model will be the best possible in that structure for the intended application. This is very different from an arbitrary experiment where it is impossible to control the model fit when a model of restricted complexity is used.

We conclude that optimal experiment design simplifies the overall system identification problem.

Identification for Control

Model-based control is one of the most important applications of system identification. Robust control ensures performance and stability in the presence of model uncertainty. However, the majority of such design methods do not employ the parametric ellipsoidal uncertainty sets resulting from standard system identification. In fact only in the last decade analysis and design tools for such type of model uncertainty have started to emerge, e.g., Raynaud et al. (2000) and Gevers et al. (2003).

The advantages of matching the identification criterion to the application have been recognized since long in this line of research. For control applications this typically implies that the identification experiment should be performed under the same closed-loop operation conditions as the controller to be designed. This was perhaps first recognized in the context of minimum variance control (see Gevers and Ljung 1986) where variance errors were the concern. Later on this was recognized to be the case also for the bias error, although here pre-filtering can be used to achieve the same objective.

To account for that the controller to be designed is not available, techniques where control and identification are iterated have been developed, cf. adaptive experiment design in section "Adaptive Experiment Design." Convergence of such schemes has been established when the true system is in the model set but has proved out of reach for models of restricted complexity.

In recent years, techniques integrating experiment design and model predictive control have started to appear. A general-purpose design criterion is used in Rathouský and Havlena (2013), while Larsson et al. (2013) uses an application-oriented criterion.

Summary and Future Directions

When there is the "luxury" to design the experiment, then this opportunity should be seized by the user. Without informative data there is little that can be done. In this exposé we have outlined the techniques that exist but also emphasized that a well-conceived experiment, reflecting the intended application, significantly can simplify the overall system identification problem.

Further developments of computational techniques are high on the agenda, e.g., how to handle

time-domain constraints and nonlinear models. To this end, developments in optimization methods are rapidly being incorporated. While, as reported in Hjalmarsson (2009), there are some results on how the identification cost depends on the performance requirements in the application, further understanding of this issue is highly desirable. Theory and further development of the emerging model predictive control schemes equipped with experiment design may very well be the direction that will have most impact in practice.

Cross-References

▶ System Identification: An Overview

Recommended Reading

A classical text on optimal experiment design is Fedorov (1972). The textbooks Goodwin and Payne (1977) and Zarrop (1979) cover this theory adapted to a dynamical system framework. A general overview is provided in Pronzato (2008). A semi-definite programming framework based on the per sample Fisher information is provided in Jansson and Hjalmarsson (2005). The least-costly framework is covered in Bombois et al. (2006). The lifting technique was introduced for input design in Manchester (2010). Details of the frequency-by-frequency design approach can be found in Ljung (1999). References to robust and adaptive experiment design can be found in Pronzato (2008) and Hjalmarsson (2009). For an account of the implications of optimal experiment design for the system identification problem as a whole, see Hjalmarsson (2009). Thorough accounts of the developments in identification for control are provided in Hjalmarsson (2005) and Gevers (2005).

Bibliography

Agüero JC, Goodwin GC (2007) Choosing between open and closed loop experiments in linear system identification. IEEE Trans Autom Control 52(8):1475–1480

Bombois X, Scorletti G, Gevers M, Van den Hof PMJ, Hildebrand R (2006) Least costly identification experiment for control. Automatica 42(10):1651–1662

Fedorov VV (1972) Theory of optimal experiments. Probability and mathematical statistics, vol 12. Academic, New York

Forssell U, Ljung L (1999) Closed-loop identification revisited. Automatica 35:1215–1241

Gevers M (2005) Identification for control: from the early achievements to the revival of experiment design. Eur J Control 11(4–5):335–352. Semi-plenary lecture at IEEE conference on decision and control – European control conference

Gevers M, Bombois X, Codrons B, Scorletti G, Anderson BDO (2003) Model validation for control and controller validation in a prediction error identification framework – part I: theory. Automatica 39(3):403–445

Gevers M, Ljung L (1986) Optimal experiment designs with respect to the intended model application. Automatica 22(5):543–554

Goodwin GC, Payne RL (1977) Dynamic system identification: experiment design and data analysis. Academic, New York

Hjalmarsson H (2005) From experiment design to closed loop control. Automatica 41(3):393–438

Hjalmarsson H (2009) System identification of complex and structured systems. Eur J Control 15(4):275–310. Plenary address. European control conference

Jansson H, Hjalmarsson H (2005) Input design via LMIs admitting frequency-wise model specifications in confidence regions. IEEE Trans Autom Control 50(10):1534–1549

Larsson CA, Hjalmarsson H, Rojas CR, Bombois X, Mesbah A, Modén P-E (2013) Model predictive control with integrated experiment design for output error systems. In: European control conference, Zurich

Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice-Hall, Englewood Cliffs

Manchester IR (2010) Input design for system identification via convex relaxation. In: 49th IEEE conference on decision and control, Atlanta, pp 2041–2046

Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. Wiley/IEEE, Hoboken/Piscataway

Pronzato L (2008) Optimal experimental design and some related control problems. Automatica 44(2):303–325

Rathouský J, Havlena V (2013) MPC-based approximate dual controller by information matrix maximization. Int J Adapt Control Signal Process 27(11):974–999

Raynaud HF, Pronzato L, Walter E (2000) Robust identification and control based on ellipsoidal parametric uncertainty descriptions. Eur J Control 6(3):245–255

Rivera DE, Lee H, Mittelmann HD, Braun MW (2009) Constrained multisine input signals for plant-friendly identification of chemical process systems. J Process Control 19(4):623–635

Rojas CR, Agüero JC, Welsh JS, Goodwin GC (2008) On the equivalence of least costly and traditional experiment design for control. Automatica 44(11):2706–2715

Rojas CR, Welsh JS, Agüero JC (2009) Fundamental limitations on the variance of parametric models. IEEE Trans Autom Control 54(5):1077–1081

Zarrop M (1979) Optimal experiment design for dynamic system identification. Lecture notes in control and information sciences, vol 21. Springer, Berlin

Explicit Model Predictive Control

Alberto Bemporad
IMT Institute for Advanced Studies Lucca,
Lucca, Italy

Abstract

Model predictive control (MPC) has been used in the process industries for more than 30 years because of its ability to control multivariable systems in an optimized way under constraints on input and output variables. Traditionally, MPC requires the solution of a quadratic program (QP) online to compute the control action, often restricting its applicability to slow processes. Explicit MPC completely removes the need for on-line solvers by precomputing the control law off-line, so that online operations reduce to a simple function evaluation. Such a function is piecewise affine in most cases, so that the MPC controller is equivalently expressed as a lookup table of linear gains, a form that is extremely easy to code, requires only basic arithmetic operations, and requires a maximum number of iterations that can be exactly computed a priori.

Keywords

Constrained control; Embedded optimization; Model predictive control; Multiparametric programming; Quadratic programming

Introduction

Model predictive control (MPC) is a well-known methodology for synthesizing feedback control laws that optimize closed-loop performance subject to prespecified operating constraints on inputs, states, and outputs (Borrelli et al. 2011; Mayne and Rawlings 2009). In MPC, the control action is obtained by solving a finite horizon open-loop optimal control problem at each sampling instant. Each optimization yields a sequence of optimal control moves, but only the first move is applied to the process: At the next time step, the computation is repeated over a shifted time horizon by taking the most recently available state information as the new initial condition of the new optimal control problem. For this reason, MPC is also called "receding horizon control." In most practical applications, MPC is based on a linear discrete-time time-invariant model of the controlled system and quadratic penalties on tracking errors and actuation efforts; in such a formulation, the optimal control problem can be recast as a quadratic programming (QP) problem, whose linear term of the cost function and right-hand side of the constraints depend on a vector of parameters that may change from one step to another (such as the current state and reference signals). To enable the implementation of MPC in real industrial products, a QP solution method must be embedded in the control hardware. The method must be fast enough to provide a solution within short sampling intervals and require simple hardware, limited memory to store the data defining the optimization problem and the code implementing the algorithm itself, a simple program code, and good worst-case estimates of the execution time to meet real-time system requirements.

Several *online* solution algorithms have been studied for embedding quadratic optimization in control hardware, such as active-set methods (Ricker 1985), interior-point methods (Wang and Boyd 2010), and fast gradient projection methods (Patrinos and Bemporad 2014). Explicit MPC takes a different approach to meet the above requirements, where multiparametric quadratic programming is proposed to pre-solve the QP

off-line, therefore converting the MPC law into a continuous and piecewise-affine function of the parameter vector (Bemporad et al. 2002b). We review the main ideas of explicit MPC in the next section, referring the reader to Alessio and Bemporad (2009) for a more complete survey paper on explicit MPC.

Model Predictive Control Problem

Consider the following finite-time optimal control problem formulation for MPC:

$$V^*(p) = \min_z \ \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \ \text{(1a)}$$

$$\text{s.t.} \ \ x_{k+1} = Ax_k + Bu_k \quad \text{(1b)}$$

$$C_x x_k + C_u u_k \leq c \quad \text{(1c)}$$

$$k = 0, \dots, N-1$$

$$C_N x_N \leq c_N \quad \text{(1d)}$$

$$x_0 = x \quad \text{(1e)}$$

where N is the prediction horizon; $x \in \mathbb{R}^m$ is the current state vector of the controlled system; $u_k \in \mathbb{R}^{n_u}$ is the vector of manipulated variables at prediction time k, $k = 0, \dots, N-1$; $z \triangleq [u_0' \cdots u_{N-1}']' \in \mathbb{R}^n$, $n \triangleq n_u N$, is the vector of decision variables to be optimized;

$$\ell(x, u) = \frac{1}{2}x'Qx + u'Ru \quad \text{(2a)}$$

$$\ell_N(x) = \frac{1}{2}x'Px \quad \text{(2b)}$$

are the stage cost and terminal cost, respectively; Q, P are symmetric and positive semidefinite matrices; and R is a symmetric and positive definite matrix.

Let $n_c \in \mathbb{N}$ be the number of constraints imposed at prediction time $k = 0, \dots, N-1$, namely, $C_x \in \mathbb{R}^{n_c \times m}$, $C_u \in \mathbb{R}^{n_c \times n_u}$, $c \in \mathbb{R}^{n_c}$, and let n_N be the number of terminal constraints, namely, $C_N \in \mathbb{R}^{n_N \times m}$, $c_N \in \mathbb{R}^{n_N}$. The total number q of linear inequality constraints imposed

in the MPC problem formulation (1) is $q = Nn_c + n_N$.

By eliminating the states $x_k = A^k x + \sum_{j=0}^{k-1} A^j Bu_{k-1-j}$ from problem (1), the optimal control problem (1) can be expressed as the convex quadratic program (QP):

$$V^\star(x) \triangleq \min_z \ \ \frac{1}{2}z'Hz + x'F'z + \frac{1}{2}x'Yx \quad \text{(3a)}$$

$$\text{s.t.} \ \ Gz \leq W + Sx \quad \text{(3b)}$$

where $H = H' \in \mathbb{R}^n$ is the Hessian matrix; $F \in \mathbb{R}^{n \times m}$ defines the linear term of the cost function; $Y \in \mathbb{R}^{m \times m}$ has no influence on the optimizer, as it only affects the optimal value of (3a); and the matrices $G \in \mathbb{R}^{q \times n}$, $S \in \mathbb{R}^{q \times m}$, $W \in \mathbb{R}^q$ define in a compact form the constraints imposed in (1). Because of the assumptions made on the weight matrices Q, R, P, matrix H is positive definite and matrix $\left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right]$ is positive semidefinite.

The MPC control law is defined by setting

$$u(x) = [I \ 0 \ \dots \ 0]z(x) \quad \text{(4)}$$

where $z(x)$ is the optimizer of the QP problem (3) for the current value of x and I is the identity matrix of dimension $n_u \times n_u$.

Multiparametric Solution

Rather than using a numerical QP solver online to compute the optimizer $z(x)$ of (3) for each given current state vector x, the basic idea of explicit MPC is to pre-solve the QP off-line for the entire set of states x (or for a convex polyhedral subset $X \subseteq \mathbb{R}^m$ of interest) to get the optimizer function z, and therefore the MPC control law u, explicitly as a function of x.

The main tool to get such an explicit solution is *multiparametric quadratic programming* (mpQP). For mpQP problems of the form (3), Bemporad et al. (2002b) proved that the optimizer function $z^* : X_f \mapsto \mathbb{R}^n$ is piecewise affine and continuous over the set X_f of parameters x for which the problem is feasible (X_f is a polyhedral set, possibly $X_f = X$) and that
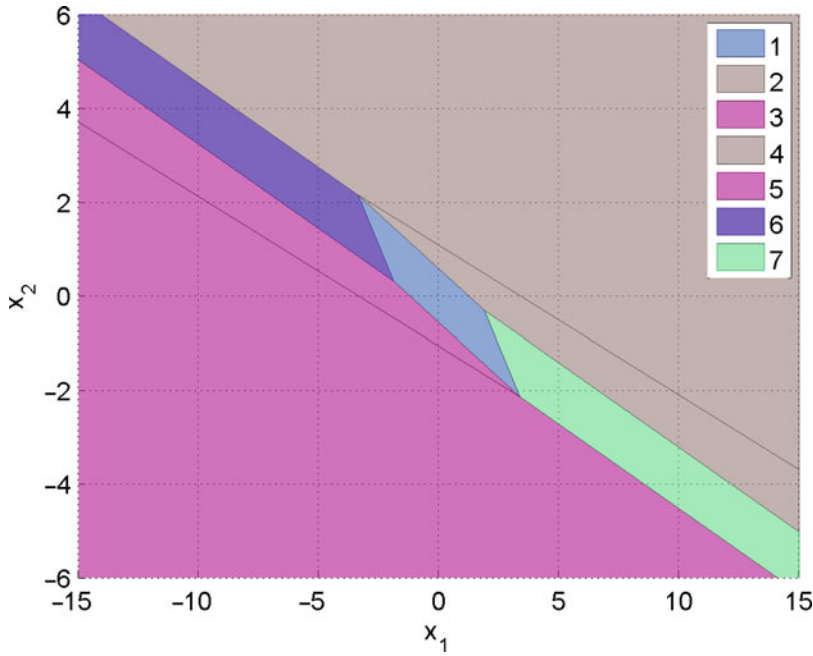
Explicit Model Predictive Control, Fig. 1 Explicit MPC solution for the double integrator example

the value function $V^* : X_f \mapsto \mathbb{R}$ associating with every $x \in X_f$ the corresponding optimal value of (3) is continuous, convex, and piecewise quadratic.

An immediate corollary is that the explicit version of the MPC control law u in (4), being the first n_u components of vector $z(x)$, is also a continuous and piecewise-affine state-feedback law defined over a partition of the set X_f of states into M polyhedral cells;

$$u(x) = \begin{cases} F_1 x + g_1 \text{ if } H_1 x \le K_1 \\ \quad\vdots \quad \vdots \\ F_M x + g_M \text{ if } H_M x \le K_M \end{cases} \quad (5)$$

An example of such a partition is depicted in Fig. 1. The explicit representation (5) has mapped the MPC law (4) into a lookup table of linear gains, meaning that for each given x, the values computed by solving the QP (3) online and those obtained by evaluating (5) are exactly the same.

Multiparametric QP Algorithms

A few algorithms have been proposed in the literature to solve the mpQP problem (3). All of them

construct the solution by exploiting the Karush-Kuhn-Tucker (KKT) conditions for optimality:

$$Hz + Fx + G'\lambda = 0 \quad (6a)$$

$$\lambda_i (G^i z - W^i - S^i x) = 0, \ \forall i = 1, \dots, q \quad (6b)$$

$$Gz \le W + Sx \quad (6c)$$

$$\lambda \ge 0 \quad (6d)$$

where $\lambda \in \mathbb{R}^q$ is the vector of Lagrange multipliers. For the strictly convex QP (3), conditions (6) are necessary and sufficient to characterize optimality.

An mpQP algorithm starts by fixing an arbitrary starting parameter vector $x_0 \in \mathbb{R}^m$ (e.g., the origin $x_0 = 0$), solving the QP (3) to get the optimal solution $z(x_0)$, and identifying the subset

$$\tilde{G}z(x) = \tilde{S}x + \tilde{W} \quad (7a)$$

of all constraints (6c) that are active at $z(x_0)$ and the remaining inactive constraints:

$$\hat{G}z(x) \le \hat{S}x + \hat{W} \quad (7b)$$

Correspondingly, in view of the complementarity condition (6b), the vector of Lagrange multipliers is split into two subvectors:

$$\tilde{\lambda}(x) \geq 0 \tag{8a}$$

$$\hat{\lambda}(x) = 0 \tag{8b}$$

We assume for simplicity that the rows of $\tilde{G}$ are linearly independent. From (6a), we have the relation

$$z(x) = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda}(x)) \tag{9}$$

that, when substituted into (7a), provides

$$\tilde{\lambda}(x) = -\tilde{M}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x) \tag{10}$$

where $\tilde{M} = \tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}$ and, by substitution in (9),

$$z(x) = H^{-1}(\tilde{M}\tilde{W} + \tilde{M}(\tilde{S} + \tilde{G}H^{-1}F)x - Fx) \tag{11}$$

The solution $z(x)$ provided by (11) is the correct one for all vectors x such that the chosen combination of active constraints remains optimal. Such all vectors x are identified by imposing constraints (7b) and (8a) on $z(x)$ and $\tilde{\lambda}(x)$, respectively, that leads to constructing the polyhedral set ("critical region"):

$$CR_0 = \{x \in \mathbb{R}^n : \tilde{\lambda}(x) \geq 0, \hat{G}z(x) \leq \hat{W} + \hat{S}x\} \tag{12}$$

Different mpQP solvers were proposed to cover the rest $X \setminus CR_0$ of the parameter set with other critical regions corresponding to new combinations of active constraints. The most efficient methods exploit the so-called "facet-to-facet" property of the multiparametric solution (Spjøtvold et al. 2006) to identify neighboring regions as in Tøndel et al. (2003a) and Baotić (2002). Alternative methods were proposed in Jones and Morari (2006), based on looking at (6) as a multiparametric linear complementarity problem, and in Patrinos and Sarimveis (2010), which provides algorithms for determining all neighboring regions even in the case the facet-to-facet property does not hold.

All methods handle the case of *degeneracy*, which may happen for some combinations of active constraints that are linearly dependent, that is, the associated matrix $\tilde{G}$ has no full row rank (in this case, $\tilde{\lambda}(x)$ may not be uniquely defined).

Extensions

The explicit approach described earlier can be extended to the following MPC setting:

$$\min_z \sum_{k=0}^{N-1} \frac{1}{2}(y_k - \mathbf{r_k})'Q_y(y_k - \mathbf{r_k}) + \frac{1}{2}\Delta u_k' R_\Delta \Delta u_k$$
$$+ (u_k - \mathbf{u_k^r})'R(u_k - \mathbf{u_k^r})' + \rho_\epsilon \epsilon^2 \tag{13a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k + B_v \mathbf{v_k} \tag{13b}$$

$$y_k = Cx_k + D_u u_k + D_v \mathbf{v_k} \tag{13c}$$

$$u_k = u_{k-1} + \Delta u_k, \ k = 0, \ldots, N-1 \tag{13d}$$

$$\Delta u_k = 0, \ k = N_u, \ldots, N-1 \tag{13e}$$

$$\mathbf{u_{min}^k} \leq u_k \leq \mathbf{u_{max}^k}, k = 0, \ldots, N_u - 1 \tag{13f}$$

$$\Delta \mathbf{u_{min}^k} \leq \Delta u_k \leq \Delta \mathbf{u_{max}^k}, \ k = 0, \ldots, N_u - 1 \tag{13g}$$

$$\mathbf{y_{min}^k} - \epsilon V_{min} \leq y_k \leq \mathbf{y_{max}^k} + \epsilon V_{max} \tag{13h}$$
$$k = 0, \ldots, N_c - 1$$

where R_Δ is a symmetric and positive definite matrix; matrices Q_y and R are symmetric and positive semidefinite; $\mathbf{v_k}$ is a vector of measured disturbances; y_k is the output vector; $\mathbf{r_k}$ its corresponding reference to be tracked; Δu_k is the vector of input increments; $\mathbf{u_k^r}$ is the input reference; $\mathbf{u_{min}^k}$, $\mathbf{u_{max}^k}$, $\Delta \mathbf{u_{min}^k}$, $\Delta \mathbf{u_{max}^k}$, $\mathbf{y_{min}^k}$, $\mathbf{y_{max}^k}$ are bounds; and N, N_u, N_c are, respectively, the prediction, control, and constraint horizons. The extra variable ϵ is introduced to soften output constraints, penalized by the (usually large) weight ρ_ϵ in the cost function (13a).

Everything marked in bold-face in (13), together with the command input u_{-1} applied at the previous sampling step and the current state x, can be treated as a parameter with respect to

which to solve the mpQP problem and obtain the explicit form of the MPC controller. For example, for a tracking problem with no anticipative action ($\mathbf{r_k} \equiv r_0, \forall k = 0, \ldots, N-1$), no measured disturbance, and fixed upper and lower bounds, the explicit solution is a continuous piecewise affine function of the parameter vector $\begin{bmatrix} x \\ r_0 \\ u_{-1} \end{bmatrix}$. Note that prediction models and/or weight matrices in (13) cannot be treated as parameters to maintain the mpQP formulation (3).

Linear MPC Based on Convex Piecewise-Affine Costs

A similar setting can be repeated for MPC problems based on linear prediction models and convex piecewise-affine costs, such as 1- and ∞-norms. In this case, the MPC problem is mapped into a multiparametric linear programming (mpLP) problem, whose solution is again continuous and piecewise-affine with respect to the vector of parameters. For details, see Bemporad et al. (2002a).

Robust MPC

Explicit solutions to min-max MPC problems that provide robustness with respect to additive and/or multiplicative unknown-but-bounded uncertainty were proposed in Bemporad et al. (2003), based on a combination of mpLP and dynamic programming. Again the solution is piecewise affine with respect to the state vector.

Hybrid MPC

An MPC formulation based on 1- or ∞-norms and hybrid dynamics expressed in mixed-logical dynamical (MLD) form can be solved explicitly by treating the optimization problem associated with MPC as a multiparametric mixed integer linear programming (mpMILP) problem. The solution is still piecewise affine but may be discontinuous, due to the presence of binary variables (Bemporad et al. 2000). A better approach based on dynamic programming combined with mpLP (or mpQP) was proposed in Borrelli et al. (2005) for hybrid systems in piecewise-affine (PWA) dynamical form and linear (or quadratic) costs.

Applicability of Explicit MPC

Complexity of the Solution

The complexity of the solution is given by the number M of regions that form the explicit solution (5), dictating the amount of memory to store the parametric solution (F_i, G_i, H_i, K_i, $i = 1, \ldots, M$), and the worst-case execution time required to compute $F_i x + G_i$ once the problem of identifying the index i of the region $\{x : H_i x \leq K_i\}$ containing the current state x is solved (which usually takes most of the time). The latter is called the "point location problem," and a few methods have been proposed to solve the problem more efficiently than searching linearly through the list of regions (see, e.g., the tree-based approach of Tøndel et al. 2003b).

An upper bound to M is 2^q, which is the number of all possible combinations of active constraints. In practice, M is much smaller than 2^q, as most combinations are never active at optimality for any of the vectors x (e.g., lower and upper limits on an actuation signal cannot be active at the same time, unless they coincide). Moreover, regions in which the first n_u component of the multiparametric solution $z(x)$ is the same can be joined together, provided that their union is a convex set (an optimal merging algorithm was proposed by Geyer et al. (2008) to get a minimal number M of partitions). Nonetheless, the complexity of the explicit MPC law typically grows exponentially with the number q of constraints. The number m of parameters is less critical and mainly affects the number of elements to be stored in memory (i.e., the number of columns of matrices F_i, H_i). The number n of free variables also affects the number M of regions, mainly because they are usually upper and lower bounded.

Computer-Aided Tools

The Model Predictive Control Toolbox (Bemporad et al. 2014) offers functions for designing explicit MPC controllers in MATLAB since 2014. Other tools exist such as the Hybrid Toolbox (Bemporad 2003) and the Multi-Parametric Toolbox (Kvasnica et al. 2006).

Summary and Future Directions

Explicit MPC is a powerful tool to convert an MPC design into an equivalent control law that can be implemented as a lookup table of linear gains. Whether the explicit form is preferable to solving the QP problem online depends on available CPU time, data memory, and program memory and other practical considerations. Although suboptimal methods have been proposed to reduce the complexity of the control law, still the explicit MPC approach remains convenient for relatively small problems (such as one or two command inputs, short control and constraint horizons, up to ten states). For larger problems, and/or problems that are linear time varying, on line QP solution methods tailored to embedded MPC may be preferable.

Cross-References

- ▶ Model-Predictive Control in Practice
- ▶ Nominal Model-Predictive Control
- ▶ Optimization Algorithms for Model Predictive Control

Recommended Reading

For getting started in explicit MPC, we recommend reading the paper by Bemporad et al. (2002b) and the survey paper Alessio and Bemporad (2009). Hands-on experience using one of the MATLAB tools listed above is also useful for fully appreciating the potentials and limitations of explicit MPC. For understanding how to program a good multiparametric QP solver, the reader is recommended to take the approach of Tøndel et al. (2003a) and Spjøtvold et al. (2006) or, in alternative, of Patrinos and Sarimveis (2010) or Jones and Morari (2006).

Bibliography

Alessio A, Bemporad A (2009) A survey on explicit model predictive control. In: Magni L, Raimondo DM, Allgower F (eds) Nonlinear model predictive control: towards new challenging applications. Lecture notes in control and information sciences, vol 384. Springer, Berlin/Heidelberg, pp 345–369

Baotić M (2002) An efficient algorithm for multi-parametric quadratic programming. Tech. Rep. AUT02-05, Automatic Control Institute, ETH, Zurich

Bemporad A (2003) Hybrid toolbox – user's guide. http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox

Bemporad A, Borrelli F, Morari M (2000) Piecewise linear optimal controllers for hybrid systems. In: Proceedings of American control conference, Chicago, pp 1190–1194

Bemporad A, Borrelli F, Morari M (2002a) Model predictive control based on linear programming – the explicit solution. IEEE Trans Autom Control 47(12):1974–1985

Bemporad A, Morari M, Dua V, Pistikopoulos E (2002b) The explicit linear quadratic regulator for constrained systems. Automatica 38(1):3–20

Bemporad A, Borrelli F, Morari M (2003) Min-max control of constrained uncertain discrete-time linear systems. IEEE Trans Autom Control 48(9):1600–1606

Bemporad A, Morari M, Ricker N (2014) Model predictive control toolbox for matlab – user's guide. The Mathworks, Inc., http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/

Borrelli F, Baotić M, Bemporad A, Morari M (2005) Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. Automatica 41(10):1709–1721

Borrelli F, Bemporad A, Morari M (2011, in press) Predictive control for linear and hybrid systems. Cambridge University Press

Geyer T, Torrisi F, Morari M (2008) Optimal complexity reduction of polyhedral piecewise affine systems. Automatica 44:1728–1740

Jones C, Morari M (2006) Multiparametric linear complementarity problems. In: Proceedings of the 45th IEEE conference on decision and control, San Diego, pp 5687–5692

Kvasnica M, Grieder P, Baotić M (2006) Multi parametric toolbox (MPT). http://control.ee.ethz.ch/~mpt/

Mayne D, Rawlings J (2009) Model predictive control: theory and design. Nob Hill Publishing, LCC, Madison

Patrinos P, Bemporad A (2014) An accelerated dual gradient-projection algorithm for embedded linear model predictive control. IEEE Trans Autom Control 59(1):18–33

Patrinos P, Sarimveis H (2010) A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings. Automatica 46(9):1405–1418

Ricker N (1985) Use of quadratic programming for constrained internal model control. Ind Eng Chem Process Des Dev 24(4):925–936

Spjøtvold J, Kerrigan E, Jones C, Tøndel P, Johansen TA (2006) On the facet-to-facet property of solutions to convex parametric quadratic programs. Automatica 42(12):2209–2214

Tøndel P, Johansen TA, Bemporad A (2003) An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica 39(3):489–497

Tøndel P, Johansen TA, Bemporad A (2003b) Evaluation of piecewise affine control via binary search tree. Automatica 39(5):945–950

Wang Y, Boyd S (2010) Fast model predictive control using online optimization. IEEE Trans Control Syst Technol 18(2):267–278

Extended Kalman Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Synonyms

EKF

Abstract

The extended Kalman filter (EKF) is the most popular estimation algorithm in practical applications. It is based on a linear approximation to the Kalman filter theory. There are thousands of variations of the basic EKF design, which are intended to mitigate the effects of nonlinearities, non-Gaussian errors, ill-conditioning of the covariance matrix and uncertainty in the parameters of the problem.

Keywords

Estimation; Nonlinear filters

The extended Kalman filter (EKF) is by far the most popular nonlinear filter in practical engineering applications. It uses a linear approximation to the nonlinear dynamics and measurements and exploits the Kalman filter theory, which is optimal for linear and Gaussian problems; Gelb (1974) is the most accessible but thorough book on the EKF. The real-time computational complexity of the EKF is rather modest; for example, one can run an EKF with high-dimensional state vectors ($d =$ several hundreds) in real time on a single microprocessor chip. The computational complexity of the EKF scales as the cube of the dimension of the state vector (d) being estimated. The EKF often gives good estimation accuracy for practical nonlinear problems, although the EKF accuracy can be very poor for difficult nonlinear non-Gaussian problems. There are many different variations of EKF algorithms, most of which are intended to improve estimation accuracy. In particular, the following types of EKFs are common in engineering practice: (1) second-order Taylor series expansion of the nonlinear functions, (2) iterated measurement updates that recompute the point at which the first order Taylor series is evaluated for a given measurement, (3) second-order iterated (i.e., combination of items 1 and 2), (4) special coordinate systems (e.g., Cartesian, polar or spherical, modified polar or spherical, principal axes of the covariance matrix ellipse, hybrid coordinates, quaternions rather than Euler angles, etc.), (5) preferred order of processing sequential scalar measurement updates, (6) decoupled or partially decoupled or quasi-decoupled covariance matrices, and many more variations. In fact, there is no such thing as "the" EKF, but rather there are thousands of different versions of the EKF. There are also many different versions of the Kalman filter itself, and all of these can be used to design EKFs as well. For example, there are many different equations to update the Kalman filter error covariance matrices with the intent of mitigating ill-conditioning and improving robustness, including (1) square-root factorization of the covariance matrix, (2) information matrix update, (3) square-root information update, (4) Joseph's robust version of the covariance matrix update, (5) at least three distinct algebraic versions of the covariance matrix update, as well as hybrids of the above.

Many of the good features of the Kalman filter are also enjoyed by the EKF, but unfortunately not all. For example, we have a very good theory of stability for the Kalman filter, but there is no theory that guarantees that an EKF will be stable in practical applications. The only method

to check whether the EKF is stable is to run Monte Carlo simulations that cover the relevant regions in state space with the relevant measurement parameters (e.g., data rate and measurement accuracy). Secondly, the Kalman filter computes the theoretical error covariance matrix, but there is no guarantee that the error covariance matrix computed by the EKF approximates the actual filter errors, but rather the EKF covariance matrix could be optimistic by orders of magnitude in real applications. Third, the numerical values of the process noise covariance matrix can be computed theoretically for the Kalman filter, but there is no guarantee that these will work well for the EKF, but rather engineers typically tune the process noise covariance matrix using Monte Carlo simulations or else use a heuristic adaptive process (e.g., IMM). All of these short-comings of the EKF compared with the Kalman filter theory are due to a myriad of practical issues, including (1) nonlinearities in the dynamics or measurements, (2) non-Gaussian measurement errors, (3) unmodeled measurement error sources (e.g., residual sensor bias), (4) unmodeled errors in the dynamics, (5) data association errors, (6) unresolved measurement data, (7) ill-conditioning of the covariance matrix, etc. The actual estimation accuracy of an EKF can only be gauged by Monte Carlo simulations over the relevant parameter space.

The actual performance of an EKF can depend crucially on the specific coordinate system that is used to represent the state vector. This is extremely well known in practical engineering applications (e.g., see Mehra 1971; Stallard 1991; Miller 1982; Markley 2007; Daum 1983; Schuster 1993). Intuitively, this is because the dynamics and measurement equations can be exactly linear in one coordinate system but not another; this is very easy to see; start with dynamics and measurements that are exactly linear in Cartesian coordinates and transform to polar coordinates and we will get highly nonlinear equations. Likewise, we can have approximately linear dynamics and measurements in a specific coordinate system but highly nonlinear equations in another coordinate system. But in theory, the optimal estimation accuracy does not depend on the coordinate system. Moreover, in math and physics, coordinate-free methods are preferred, owing to their greater generality and simplicity and power. The physics does not depend on the specific coordinate system; this is essentially a definition of what "physics" means, and it has resulted in great progress in physics over the last few hundred years (e.g., general relativity, gauge invariance in quantum field theory, Lorentz invariance in special relativity, as well as a host of conservation laws in classical mechanics that are explained by Noether's theorem which relates invariance to conserved quantities). Similarly in math, coordinate-free methods have been the royal road to progress over the last 100 years but not so for practical engineering of EKFs, because EKFs are approximations rather than being exact, and the accuracy of the EKF approximation depends crucially on the specific coordinate system used. Moreover, the effect of ill-conditioning of the covariance matrices in EKFs depends crucially on the specific coordinate system used in the computer; for example, if we could compute the EKF in principal coordinates, then the covariance matrices would be diagonal, and there would be no effect of ill-conditioning, despite enormous condition numbers of the covariance matrices. Surprisingly, these two simple points about coordinate systems are still not well understood by many researchers in nonlinear filtering.
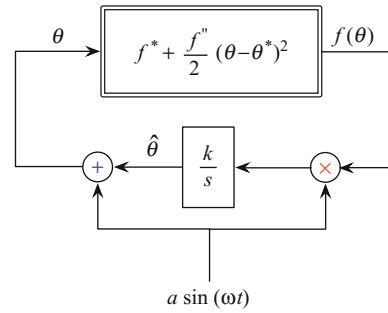
Cross-References

▶ Estimation, Survey on
▶ Kalman Filters
▶ Nonlinear Filters
▶ Particle Filters

Bibliography

Crisan D, Rozovskii B (eds) (2011) The Oxford handbook of nonlinear filtering. Oxford University Press, Oxford/New York

Daum FE, Fitzgerald RJ (1983) Decoupled Kalman filters for phased array radar tracking. IEEE Trans Autom Control 28:269–283

Gelb A et al (1974) Applied optimal estimation. MIT, Cambridge

Markley FL, Crassidis JL, Cheng Y (2007) Nonlinear attitude filtering methods. AIAA J 30:12–28

Mehra R (1971) A comparison of several nonlinear filters for reentry vehical tracking. IEEE Trans Autom Control 16:307–310

Miller KS, Leskiw D (1982) Nonlinear observations with radar measurements. IEEE Trans Aerosp Electron Syst 2:192–200

Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter. Artech House, Boston

Schuster MD (1993) A survey of attitude representations. J Astronaut Sci 41:439–517

Sorenson H (ed) (1985) Kalman filtering: theory and application. IEEE, New York

Stallard T (1991) Angle-only tracking filter in modified spherical coordinates. AIAA J Guid 14:694–696

Tanizaki H (1996) Nonlinear filters, 2nd edn. Springer, Berlin/New York

Extremum Seeking Control

Miroslav Krstic
Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Abstract

Extremum seeking (ES) is a method for real-time non-model-based optimization. Though ES was invented in 1922, the "turn of the twenty-first century" has been its golden age, both in terms of the development of theory and in terms of its adoption in industry and in fields outside of control engineering. This entry overviews basic gradient- and Newton-based versions of extremum seeking with periodic and stochastic perturbation signals.

Keywords

Gradient climbing; Newton's method

The Basic Idea of Extremum Seeking

Many versions of extremum seeking exist, with various approaches to their stability study (Krstic and Wang 2000; Liu and Krstic 2012; Tan et al.

Extremum Seeking Control, Fig. 1 The simplest perturbation-based extremum seeking scheme for a quadratic single-input map $f(\theta) = f^* + \dfrac{f''}{2}(\theta - \theta^*)^2$, where f^*, f'', θ^* are all unknown. The user has to only know the sign of f'', namely, whether the quadratic map has a maximum or a minimum, and has to choose the adaptation gain k such that $\mathrm{sgn}\,k = -\mathrm{sgn}\,f''$. The user has to also choose the frequency ω as relatively large compared to a, k, and f''

2006). The most common version employs perturbation signals for the purpose of estimating the gradient of the unknown map that is being optimized. To understand the basic idea of extremum seeking, it is best to first consider the case of a static single-input map of the quadratic form, as shown in Fig. 1.

Three different thetas appear in Fig. 1: θ^* is the unknown optimizer of the map, $\hat{\theta}(t)$ is the real-time estimate of θ^*, and $\theta(t)$ is the actual input into the map. The actual input $\theta(t)$ is based on the estimate $\hat{\theta}(t)$ but is perturbed by the signal $a\sin(\omega t)$ for the purpose of estimating the unknown gradient $f'' \cdot (\theta - \theta^*)$ of the map $f(\theta)$. The sinusoid is only one choice for a perturbation signal – many other perturbations, from square waves to stochastic noise, can be used in lieu of sinusoids, provided they are of zero mean. The estimate $\hat{\theta}(t)$ is generated with the integrator k/s with the adaptation gain k controlling the speed of estimation.

The ES algorithm is successful if the error between the estimate $\hat{\theta}(t)$ and the unknown θ^*, namely, the signal

$$\tilde{\theta}(t) = \hat{\theta}(t) - \theta^* \tag{1}$$

converges towards zero. Based on Fig. 1, the estimate is governed by the differential equation $\dot{\hat{\theta}} = k \sin(\omega t) f(\theta)$, which means that the estimation error is governed by

$$\frac{\mathrm{d}\tilde{\theta}}{\mathrm{d}t} = ka \sin(\omega t) \left[f^* + \frac{f''}{2} \left(\tilde{\theta} + a \sin(\omega t) \right)^2 \right] \tag{2}$$

Expanding the right-hand side, one obtains

$$\frac{\mathrm{d}\tilde{\theta}(t)}{\mathrm{d}t} = kaf^* \underbrace{\sin(\omega t)}_{\text{mean}=0} + ka^3 \frac{f''}{2} \underbrace{\sin^3(\omega t)}_{\text{mean}=0}$$

$$+ ka \frac{f''}{2} \underbrace{\sin(\omega t)}_{\text{fast, mean}=0} \underbrace{\tilde{\theta}(t)^2}_{\text{slow}}$$

$$+ ka^2 f'' \underbrace{\sin^2(\omega t)}_{\text{fast, mean}=1/2} \underbrace{\tilde{\theta}(t)}_{\text{slow}} \tag{3}$$

A theoretically rigorous time-averaging procedure allows to replace the above sinusoidal signals by their means, yielding the "average system"

$$\frac{\mathrm{d}\tilde{\theta}_{\text{ave}}}{\mathrm{d}t} = \overbrace{\frac{kf'' a^2}{2}}^{<0} \tilde{\theta}_{\text{ave}}, \tag{4}$$

which is exponentially stable. The averaging theory guarantees that there exists sufficiently large ω such that, if the initial estimate $\hat{\theta}(0)$ is sufficiently close to the unknown θ^*,

$$|\theta(t) - \theta^*| \le |\theta(0) - \theta^*| e^{\frac{kf''a^2}{2}t} + O\left(\frac{1}{\omega}\right)$$

$$+ a, \qquad \forall t \ge 0. \tag{5}$$

For the user, the inequality (5) guarantees that, if a is chosen small and ω is chosen large, the input $\theta(t)$ exponentially converges to a small interval around the unknown θ^* and, consequently, the output $f(\theta(t))$ converges to the vicinity of the optimal output f^*.
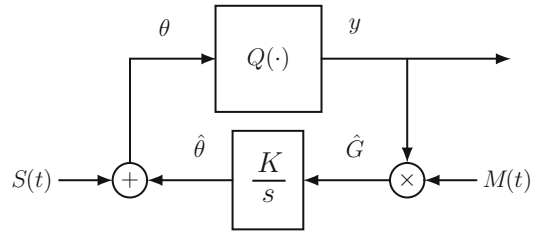
Extremum Seeking Control, Fig. 2 Extremum seeking algorithm for a multivariable map $y = Q(\theta)$, where θ is the input vector $\theta = [\theta_1, \theta_2, \cdots, \theta_n]^T$. The algorithm employs the additive perturbation vector signal $S(t)$ given in (6) and the multiplicative demodulation vector signal $M(t)$ given in (7)

ES for Multivariable Static Maps

For static maps, ES extends in a straightforward manner from the single-input case shown in Fig. 1 to the multi-input case shown in Fig. 2.

The algorithm measures the scalar signal $y(t) = Q(\theta(t))$, where $Q(\cdot)$ is an unknown map whose input is the vector $\theta = [\theta_1, \theta_2, \cdots, \theta_n]^T$. The gradient is estimated with the help of the signals

$$S(t) = \begin{bmatrix} a_1 \sin(\omega_1 t) & \cdots & a_n \sin(\omega_n t) \end{bmatrix}^T \tag{6}$$

$$M(t) = \begin{bmatrix} \dfrac{2}{a_1} \sin(\omega_1 t) & \cdots & \dfrac{2}{a_n} \sin(\omega_n t) \end{bmatrix}^T \tag{7}$$

with nonzero perturbation amplitudes a_i and with a gain matrix K that is diagonal. To guarantee convergence, the user should choose $\omega_i \ne \omega_j$. This is a key condition that differentiates the multi-input case from the single-input case. In addition, for simplicity in the convergence analysis, the user should choose ω_i/ω_j as rational and $\omega_i + \omega_j \ne \omega_k$ for distinct i, j, and k.

If the unknown map is quadratic, namely, $Q(\theta) = Q^* + \frac{1}{2}(\theta - \theta^*)^T H(\theta - \theta^*)$, the averaged system is

$$\dot{\tilde{\theta}}_{\text{ave}} = KH\tilde{\theta}_{\text{ave}}, \qquad H = \text{Hessian}. \tag{8}$$
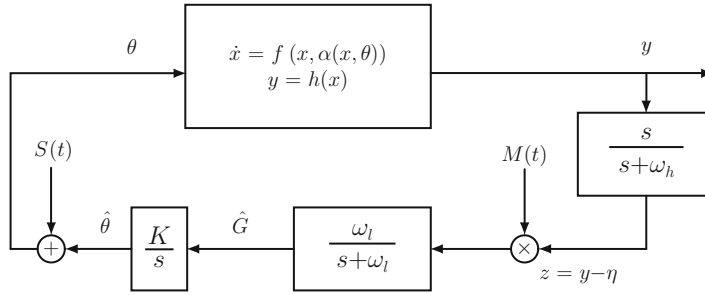
Extremum Seeking Control, Fig. 3 The ES algorithm in the presence of dynamics with an equilibrium map $\theta \mapsto y$ that satisfies the same conditions as in the static case. If the dynamics are stable and the user employs parameters in the ES algorithm that make the algorithm dynamics slower than the dynamics of the plant, convergence is guaranteed (at least locally). The two filters are useful in the implementation to reduce the adverse effect of the perturbation signals on asymptotic performance but are not needed in the stability analysis

If, for example, the map $Q(\cdot)$ has a maximum that is locally quadratic (which implies $H = H^T < 0$) and if the user chooses the elements of the diagonal gain matrix K as positive, the ES algorithm is guaranteed to be locally convergent. However, the convergence rate depends on the unknown Hessian H. This weakness of the gradient-based ES algorithm is removed with the Newton-based ES algorithm.

A stochastic version of the algorithm in Fig. 2 also exists, in which $S(t)$ and $M(t)$ are replaced by

$$S(\eta(t)) = [a_1 \sin(\eta_1(t)), \ldots, a_n \sin(\eta_n(t))]^T,$$

$$\tag{9}$$

$$M(\eta(t)) = \left[\frac{2}{a_1(1 - e^{-q_1^2})} \sin(\eta_1(t)), \ldots, \right.$$

$$\left. \frac{2}{a_n(1 - e^{-q_n^2})} \sin(\eta_n(t)) \right]^T \tag{10}$$

where $\eta_i = \dfrac{q_i \sqrt{\varepsilon_i}}{\varepsilon_i s + 1} [\dot{W}_i]$ and $\dot{W}_i$ are independent unity-intensity white noise processes.

ES for Dynamic Systems

ES extends in a relatively straightforward manner from static maps to dynamic systems, provided the dynamics are stable and the algorithm's parameters are chosen so that the algorithm's dynamics are slower than those of the plant. The algorithm is shown in Fig. 3.

The technical conditions for convergence in the presence of dynamics are that the equilibria $x = l(\theta)$ of the system $\dot{x} = f(x, \alpha(x, \theta))$, where $\alpha(x, \theta)$ is the control law of an internal feedback loop, are locally exponentially stable uniformly in θ and that, given the output map $y = h(x)$, there exists at least one $\theta^* \in \mathbb{R}^n$ such that $\dfrac{\partial}{\partial \theta}(h \circ l)(\theta^*) = 0$ and $\dfrac{\partial^2}{\partial \theta^2}(h \circ l)(\theta^*) = H < 0$, $H = H^T$.

The stability analysis in the presence of dynamics employs both averaging and singular perturbations, in a specific order. The design guidelines for the selection of the algorithm's parameters follow the analysis. Though the guidelines are too lengthy to state here, they ensure that the plant's dynamics are on a fast time scale, the perturbations are on a medium time scale, and the ES algorithm is on a slow time scale.

Newton ES Algorithm for Static Map

A Newton version of the ES algorithm, shown in Fig. 4, ensures that the convergence rate be user assignable, rather than being dependent on the unknown Hessian of the map.

The elements of the demodulating matrix $N(t)$ for generating the estimate of the Hessian are given by
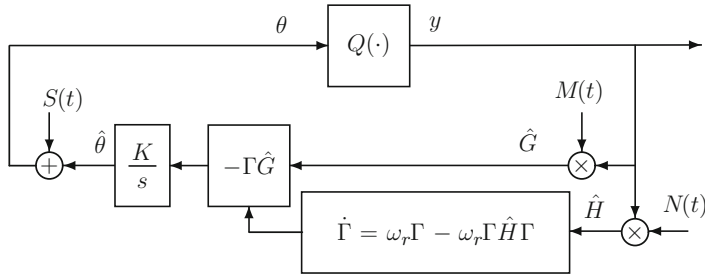
Extremum Seeking Control, Fig. 4 A Newton-based ES algorithm for a static map. The multiplicative excitation $N(t)$ helps generate the estimate of Hessian $\dfrac{\partial^2 Q(\theta)}{\partial \theta^2}$

as $\hat{H}(t) = N(t)y(t)$. The Riccati matrix differential equation $\Gamma(t)$ generates an estimate of the Hessian's *inverse* matrix, avoiding matrix inversions of Hessian estimates that may be singular during the transient

$$N_{ii}(t) = \frac{16}{a_i^2}\Big(\sin^2(\omega_i t) - \frac{1}{2}\Big),$$

$$N_{ij}(t) = \frac{4}{a_i a_j}\sin(\omega_i t)\sin(\omega_j t) \qquad (11)$$

For a quadratic map, the averaged system in error variables $\tilde{\theta} = \hat{\theta} - \theta^*$, $\tilde{\Gamma} = \Gamma - H^{-1}$ is

$$\frac{d\tilde{\theta}^{\text{ave}}}{dt} = -K\tilde{\theta}^{\text{ave}} - K\underbrace{\tilde{\Gamma}^{\text{ave}}H\tilde{\theta}^{\text{ave}}}_{\text{quadratic}},$$

$$\frac{d\tilde{\Gamma}^{\text{ave}}}{dt} = -\omega_r\tilde{\Gamma}^{\text{ave}} - \omega_r\underbrace{\tilde{\Gamma}^{\text{ave}}H\tilde{\Gamma}^{\text{ave}}}_{\text{quadratic}}. \quad (12)$$

Since the eigenvalues are determined by K and ω_r and are therefore independent of the unknown H, the (local) convergence rate is user assignable.

Further Reading on Extremum Seeking

Since the publication of the first proof of stability of extremum seeking (Krstic and Wang 2000), thousands of papers have been published

on this topic, presenting further theoretical developments and applications of ES. A proof that expands the validity of extremum seeking from local to global stability was published in Tan et al. (2006). The book Liu and Krstic (2012) presents stochastic versions of the algorithms in this entry, where the sinusoids are replaced by filtered white noise perturbation signals.

Cross-References

▸ Adaptive Control, Overview
▸ Optimal Deployment and Spatial Coverage

Bibliography

Krstic M, Wang HH (2000) Stability of extremum seeking feedback for general dynamic systems. Automatica 36:595–601

Liu S-J, Krstic M (2012) Stochastic averaging and stochastic extremum seeking. Springer, London/New York

Tan Y, Nesic D, Mareels I (2006) On non-local stability properties of extremum seeking control. Automatica 42:889–903

F

Fault Detection and Diagnosis

Janos Gertler
George Mason University, Fairfax, VA, USA

Synonyms

FDD

Abstract

The fundamental concepts and methods of fault detection and diagnosis are reviewed. Faults are defined and classified as additive or multiplicative. The model-free approach of alarm systems is described and critiqued. Residual generation, using the mathematical model of the plant, is introduced. The propagation of additive and multiplicative faults to the residuals is discussed, followed by a review of the effect of disturbances, noise, and model errors. Enhanced residuals (structured and directional) are introduced. The main residual generation techniques are briefly described, including direct consistency relations, parity space, and diagnostic observers. Principal component analysis and its application to fault detection and diagnosis are outlined. The article closes with some thoughts about future directions.

Keywords

Consistency relations; Diagnostic observers; Fault detection; Fault diagnosis; Parity space; Principal component analysis; Residual generation

Introduction

Faults are malfunctions of various elements of technical systems. Extreme cases of faults, called *failures*, are catastrophic breakdowns of the same. The technical systems (*the plant*) we are concerned with range from complex production systems (chemical plants, oil refineries, power stations) through major transportation equipment (airplanes, ships) to consumer machines (automobiles, home-heating systems, etc.). The faults may affect various parts of the main technical system (motors, pumps, storage tanks, pipelines) or devices interfacing the main technical system with computers providing for control, monitoring, and operator information. These latter include *sensors* (measuring devices) and *actuators* (devices acting on the process, such as valves).

The objective of *fault detection* is to determine and signal if there is a fault anywhere in the system. *Fault diagnosis* is aimed at providing more specific information about the fault; *fault isolation* is to pinpoint at the component(s) (sensors, actuators, or plant components) where the fault is located, while *fault identification* is to determine (estimate) the size of the fault and, in

some cases, the time of its arrival. With the ubiquitous presence of the computer, fault detection and diagnosis (FDD) is, in general, a function of the computer interfaced to the plant.

The simplest approaches to FDD consist of comparing individual plant measurements to preset limits, without utilizing any knowledge of the plant model (*limit checking* or *alarm systems*). More sophisticated techniques rely on an explicit mathematical model of the plant. They compare plant measurements to estimates obtained, from other measurements, by the model; any discrepancy may be an indication of faults. Another class of techniques (generally but incorrectly called "*data driven*"), most notably principal component analysis (PCA), include the estimation of an implicit model, from empirical plant data, and then use this in ways similar to the model-based methods. These approaches will be described in more detail in the sequel.

Alarm Systems

Alarm systems rely on the comparison of individual plant measurements to their respective limits. The limits may be two or one sided (upper and lower limit or upper limit only) and may have one or two levels (preliminary and full alarm). Momentary comparisons may be extended to include trend checks. Alarm systems are relatively simple but suffer from two major shortcomings:

– They have very limited fault specificity. A variable exceeding its limit is not a fault but a symptom of faults. A single-component fault may cause alarm on many variables and a particular alarm may be due to various component faults.
– They have limited fault sensitivity. What is "normal" for a plant output variable depends on the value of the plant inputs. Such relationship, however, cannot be considered without a plant model; therefore, the alarm thresholds need to be set conservatively high.

Because of their simplicity, and in spite of the above shortcomings, alarm systems are widely used in industrial applications.

Model-Based FDD Concepts

Model-based methods utilize an explicit mathematical model of the plant. Such model is obtained usually from empirical plant data by systems identification methods or, exceptionally, from the "first principles" understanding of the plant. Model building, though critical to the success of model-based FDD, is usually not considered part of the FDD effort. The models may be linear or nonlinear, static or dynamic, and continuous or discrete time. In FDD, most frequently linear discrete-time dynamic models are used.

The fundamental idea of model-based FDD is the comparison of measured plant outputs to their estimates, obtained, via the mathematical model, from measured or actuated plant inputs (Fig. 1). Any discrepancy is (at least ideally) an indication that a fault (or faults) is (are) present in the system. Mathematically, the difference between the measured output $y_i(t)$ and its estimate $y_i{}^{\wedge}(t)$ is a (primary) residual (Willsky 1976):

$$e_i(t) = y_i(t) - y_i{}^{\wedge}(t)$$

In general, residuals are quantities that are zero in the absence of faults and nonzero in their presence.

Unfortunately, it is not only the faults that can make the residuals nonzero. Usually, the plant is subject to disturbances (unmeasured determin-
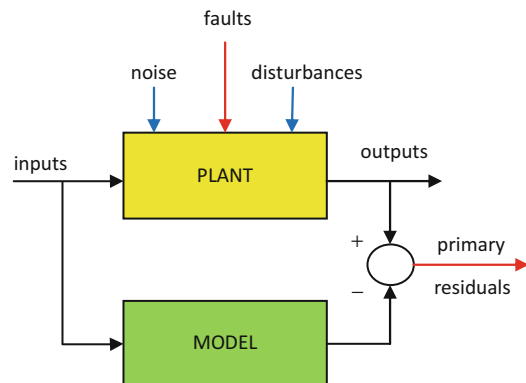
Fault Detection and Diagnosis, Fig. 1 Analytical redundancy

istic inputs) and noise (unmeasured random inputs) (Fig. 1). In addition, and most importantly, model-based FDD is subject to model errors (due either to initial inaccuracies in model building or to changes in the physical plan). The FDD algorithm should be designed, as much as possible, to be insensitive to noise and "*robust*" in face of disturbances and model errors.

Additive and multiplicative faults. Depending on the way they appear in the system equations, faults may be additive or multiplicative. Additive faults are sensor and actuator biases, leaks in the plant, etc. Multiplicative faults are changes in the plant parameters. In the following input-output relationship, $\mathbf{u}(t)$ is the vector of observed (measured or commanded) plant inputs, $\mathbf{y}(t)$ is the vector of measured plant outputs, and $\mathbf{p}(t)$ is the vector of additive faults and t is the discrete time. $\mathbf{M}(q)$ and $\mathbf{S}(q)$ are transfer function matrices in the shift operator q, and $\boldsymbol{\theta}$ is the vector of plant parameters. Then,

$$\mathbf{y}(t) = \mathbf{M}(q, \boldsymbol{\theta})\mathbf{u}(t) + \mathbf{S}(q, \boldsymbol{\theta})\mathbf{p}(t)$$

The ("primary") residual vector $\mathbf{e}(t)$, in response to additive faults, is

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{M}(q, \boldsymbol{\theta})\mathbf{u}(t) = \mathbf{S}(q, \boldsymbol{\theta})\mathbf{p}(t)$$

If there are multiplicative faults, then $\boldsymbol{\theta} = \boldsymbol{\theta}^{\circ} + \boldsymbol{\Delta\theta}$, where $\boldsymbol{\theta}^{\circ}$ is the nominal parameter vector and $\boldsymbol{\Delta\theta}$ is its change (the parametric fault); now and the residual vector $\mathbf{e}(t)$ is (Gertler 1998)

$$\begin{aligned}\mathbf{e}(t) &= \mathbf{y}(t) - \mathbf{M}(q, \boldsymbol{\theta}^{\circ})\mathbf{u}(t) \\ &= \Sigma_j(\partial\mathbf{M}(q, \boldsymbol{\theta})/\partial\theta_j)\mathbf{u}(t)\,\Delta\theta_j\end{aligned}$$

Enhanced residuals. To facilitate the isolation of faults, the primary residuals $\mathbf{e}(t)$ are subject to some enhancement manipulation. The three widely used enhancement techniques are:

– *Structured residuals*, whereas each residual is selectively sensitive to a subset of faults, resulting in a fault-specific set of zero/nonzero residuals upon a particular fault (fault codes)
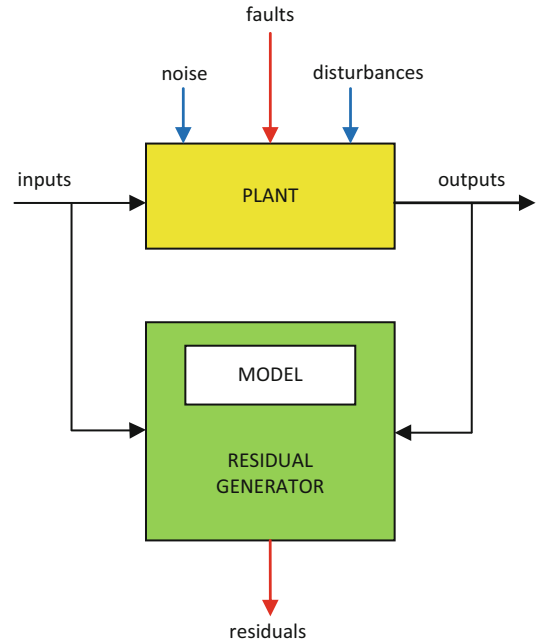
Fault Detection and Diagnosis, Fig. 2 Generating model-based residuals

– *Directional residuals*, whereas the residual vector maintains a fault-specific direction in response to each particular fault
– *Diagonal residuals*, whereas each residual responds only to a particular fault

Residual generators take the input and output observations from the plant and generate enhanced residuals by one of the above schemes, utilizing the mathematical model of the plant (Fig. 2).

Dealing with noise. Noise is practically unavoidable in physical systems. In FDD, basically two steps may be taken to reduce the effect of noise:

– *Residual filtering*. This can be achieved by basing decisions on moving averages of the residuals or by applying explicit low-pass filters to the residuals or by designing the residual generators in such a way that they have built-in low-pass behavior.
– *Statistical testing of the residuals*. Structured residuals are tested individually; each scalar residual is then represented by a Boolean 1 or 0, depending on the outcome of the

test. Directional residuals are tested as vectors against multivariable distributions. The test thresholds are determined either theoretically, using assumptions for the source noise, or empirically based on measurements from fault-free operating conditions.

Dealing with disturbances. Additive disturbances are unmeasurable inputs. If the disturbance-to-output transfer function (or equivalent state-space representation) is known, then it is possible to design residuals that are completely decoupled from (insensitive to) those disturbances. However, the FDD algorithm is subject to a certain degree of "design freedom," defined by the number of outputs in the physical system; disturbance decoupling is competing for this freedom with fault isolation enhancement. If there are too many disturbances, or if their path to the outputs is unknown, then only approximate decoupling is possible, making FDD also approximate, usually designed to optimize some (H-infinity) performance index.

Dealing with model errors. Model errors are also unavoidable in most practical situations. This is the most serious obstacle in the application of model-based FDD techniques. In some very special cases, uncertainty of a particular plant parameter may be handled as a "multiplicative disturbance," and residuals designed to be explicitly decoupled from it. In general, however, only approximate solutions are possible, reducing the residuals' sensitivity to modeling errors, at the expense of also reducing their sensitivity to faults. Design methods utilizing some optimization techniques, mostly based on H-infinity or similar performance indices, are available in the literature (Edelmayer et al. 1994).

Residual Generation Methods

For linear dynamic systems, provided exact (non-approximate) solution is possible, there are three major techniques to design residual generators: (i) direct consistency (parity) relations, (ii) parity space, and (iii) diagnostic observers. We will briefly introduce the three methods, for discrete-time plant models and additive faults. Note that though they look formally different, if designed for the same plant under the same design conditions, the three methods yield identical residuals (Gertler 1991).

Direct consistency (parity) relations (Gertler 1998)**.** The input-output model of the plant is utilized directly in the design. The enhanced residuals are obtained from the primary residuals by a transformation $\mathbf{W}(q)$:

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{W}(q)\mathbf{e}(t) = \mathbf{W}(q)[\mathbf{y}(t) - \mathbf{M}(q)\mathbf{u}(t)] \\ &= \mathbf{W}(q)\mathbf{S}(q)\mathbf{p}(t) \end{aligned}$$

The desired behavior of the residuals is specified as $\mathbf{r}(t) = \mathbf{Z}(q)\mathbf{p}(t)$, where the specification $\mathbf{Z}(q)$ contains the basic residual properties (structure or directions) plus the residual dynamics. The resulting design condition is $\mathbf{W}(q)\mathbf{S}(q) = \mathbf{Z}(q)$. If the $\mathbf{S}(q)$ matrix is square, what is usually the case (Gertler 1998), then this can be solved for $\mathbf{W}(q)$ by direct inversion. The residual generator has to be causal and stable; this can always be achieved by the appropriate modification of the dynamics in $\mathbf{Z}(q)$.

Parity space (Chow and Willsky 1984). This method, also known as the "Chow-Willsky scheme," relies on the state-space description of the system:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) + \mathbf{E}\,\mathbf{p}(t) \\ \mathbf{y}(t) &= \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{u}(t) + \mathbf{F}\,\mathbf{p}(t) \end{aligned}$$

Stacking n consecutive output vectors $\mathbf{y}(t)$ (where n is the order of the model), and chain-substituting the state $\mathbf{x}(t)$, yields the equation

$$\mathbf{Y}(t) = \mathbf{J}\,\mathbf{x}(t-n) + \mathbf{K}\,\mathbf{U}(t) + \mathbf{L}\,\mathbf{P}(t)$$

where $\mathbf{Y}(t), \mathbf{U}(t)$, and $\mathbf{P}(t)$ are stacked vectors and $\mathbf{J}, \mathbf{K}$, and $\mathbf{L}$ are hyper-matrices composed of the $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}$ matrices. Now

$$\mathbf{E}^*(t) = \mathbf{Y}(t) - \mathbf{K}\,\mathbf{U}(t) = \mathbf{L}\,\mathbf{P}(t) + \mathbf{J}\,\mathbf{x}(t-n)$$

would be a stacked vector of primary residuals, was it not for the presence of the inaccessible initial state $\mathbf{x}(t-n)$. To obtain true residuals, a transformation $r_i(t) = \mathbf{w}_i \mathbf{E}^*(t)$ is necessary, so that $\mathbf{w}_i \mathbf{J} = 0$. Any vector $\mathbf{w}_i$ satisfying this orthogonality condition is a *parity vector*, together spanning the *parity space*. Any parity vector yields a true residual $r_i(t)$; they can be so chosen that a set of residuals possesses structured behavior.

Diagnostic observers. Various observer schemes have been extensively investigated as possible residual generator algorithms. The basic full-order *Luenberger observer* (assuming $\mathbf{D} = \mathbf{0}$) is

$$\mathbf{x}^{(t+1)} = \mathbf{A}\,\mathbf{x}^{(t)} + \mathbf{B}\,\mathbf{u}(t) + \mathbf{K}\,\mathbf{e}(t)$$

where $\mathbf{K}$ is the observer gain matrix and

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{C}\,\mathbf{x}^{(t)}$$

is the innovation vector. If the observer is stable then, apart from the start-up transient of the observer, the innovation qualifies as the primary residual. The gain matrix $\mathbf{K}$ is the major design parameter; it is chosen to place the observer poles, thus achieving stability and desired dynamic behavior (e.g., noise suppression). The remaining design freedom can be utilized to influence residual properties. The latter are further affected by the transformation $\mathbf{r}(t) = \mathbf{H}\,\mathbf{e}(t)$, where the $\mathbf{H}$ matrix is an additional design parameter. Diagnostic observers can be designed for both structured and directional residuals (Chen and Patton 1999; White and Speyer 1987). Other observer schemes, most notably the *unknown input observer*, have also been proposed (Frank and Wunnenberg 1989). Because of their complexity, the detailed design procedures of diagnostic observers go beyond the scope of this entry.

Principal Component Analysis

Principal component analysis is extensively used in the monitoring of complex plants with hundreds of variables because, by revealing linear relations among the variables, it significantly reduces the dimensionality of the plant model (Kresta et al. 1991). The application of PCA for FDD implies two phases. In the training phase, an implicit plant model is created from empirical plant data. In the monitoring phase, this model is used for FDD.

Training data (measured inputs and outputs) are collected from the plant during fault-free operation. The covariance matrix of the data is formed and its eigenstructure obtained. Due to linear relations among the data, some of the eigenvalues will be zero (or near zero, in the presence of noise). The eigenvectors belonging to the nonzero eigenvalues form the *data space*, where the fault-free data exist, while those belonging to the zero eigenvalues form the *residual space*.

It is the residual space that is utilized for FDD. The projection of a measurement vector onto the residual space is the (primary) residual. A statistical test on its size leads to a detection decision (the absence or presence of faults). A threshold test is necessary because noise also causes nonzero residuals. An analysis of the eigenvectors spanning the residual space shows how the various faults propagate to the primary residual. This allows for the design of residual manipulations yielding structured or directional residuals, just like in the FDD methods based on exact models (Gertler et al. 1999).

The procedure as described above applies to sensor and actuator faults; inclusion of plant faults requires extra effort (and experiments). Also, PCA is primarily meant for static models. Its extension to discrete-time dynamic models is straightforward, but it increases the size of the model, proportionally to the dynamic order of the model.

Summary and Future Directions

Fault detection and diagnosis is today a mature field of systems and control engineering. There is a very significant level of activity, as measured in published papers and conference contributions, but much of this (in the opinion of this author)

is just minor refinements of earlier results. This applies particularly to the long ongoing quest to create "robust" FDD algorithms, especially in the face of model errors.

There are still open challenges in a couple of areas, most notably extensions to various nonlinear or parameter varying problems. Another open and active area, of great practical importance, is FDD in networked control systems. What is really of the greatest interest, though, is the application of the wealth of available theoretical results and design methods to real-life problems; there has recently been some visible progress here, a most welcome development.

Cross-References

- ▶ Controller Performance Monitoring
- ▶ Diagnosis of Discrete Event Systems
- ▶ Fault-Tolerant Control
- ▶ Multiscale Multivariate Statistical Process Control
- ▶ Observers for Nonlinear Systems
- ▶ Robust Fault Diagnosis and Control
- ▶ Statistical Process Control in Manufacturing

Bibliography

Chen J, Patton RJ (1999) Robust model-based fault diagnosis for dynamic systems. Kluwer, Boston/Dordrecht/Amsterdam

Chow EJ, Willsky AS (1984) Analytical redundancy and the design of robust failure detection systems. IEEE Trans Autom Control AC-29:603–614

Edelmayer A, Bokor J, Keviczky L (1994) An H-infinity filtering approach to robust detection of failures in dynamic systems. In: 33rd IEEE conference on decision and control, Lake Buena Vista

Frank PM, Wunnenberg J (1989) Robust fault diagnosis using unknown input observer schemes. In: Patton R, Frank P, Clark R (eds) Fault diagnosis in dynamic systems. Prentice Hall, Upper Saddle River

Gertler J (1991) A survey of analytical redundancy methods in fault detection and isolation. Plenary paper, IFAC Safeprocess Symposium, Baden-Baden

Gertler J (1998) Fault detection and diagnosis in engineering systems. Marcel Dekker, New York

Gertler J, Li W, Huang Y, McAvoy T (1999) Isolation enhanced principal component analysis. AIChE J 45:323–334

Isermann R (1984) Process fault detection based on modeling and estimation methods. Automatica 20:387–404

Kresta JV, MacGregor JF, Marlin TE (1991) Multivariate statistical monitoring of processes. Can J Chem Eng 69:35–47

White JE, Speyer JL (1987) Detection filter design: spectral theory and algorithm. IEEE Trans Autom Control AC-32:593–603

Willsky AS (1976) A survey of design methods for failure detection in dynamic systems. Automatica 12:601–611

Fault-Tolerant Control

Ron J. Patton
School of Engineering, University of Hull, Hull, UK

Synonyms

FTC

Abstract

A closed-loop control system for an engineering process may have unsatisfactory performance or even instability when faults occur in actuators, sensors, or other process components. Fault-tolerant control (FTC) involves the development and design of special controllers that are capable of tolerating the actuator, sensor, and process faults while still maintaining desirable and robust performance and stability properties. FTC designs involve knowledge of the nature and/or occurrence of faults in the closed-loop system either implicitly or explicitly using methods of fault detection and isolation (FDI), fault detection and diagnosis (FDD), or fault estimation (FE). FTC controllers are reconfigured or restructured using FDI/FDD information so that the effects of the faults are reduced or eliminated within each feedback loop in *active* or *passive* approaches or compensated in each control-loop using FE methods. A non-mathematical outline of the essential features of FTC systems is given with important definitions and a classification of FTC systems

into either active/passive approaches with examples of some well-known strategies.

Keywords

Active FTC; Fault accommodation; Fault detection and diagnosis (FDD); Fault detection and isolation (FDI); Fault estimation (FE); Fault-tolerant control; Passive FTC; Reconfigurable control

Introduction

The complexity of modern engineering systems has led to strong demands for enhanced control system reliability, safety, and green operation in the presence of even minor anomalies. There is a growing need not only to determine the onset and development of process faults before they become serious but also to adaptively compensate for their effects in the closed-loop system or using hardware redundancy to replace faulty components by duplicate and fault-free alternatives. The title "failure tolerant control" was given by Eterno et al. (1985) working on a reconfigurable flight control study defining the meaning of control system tolerance to failures or faults. The word "failure" is used when a fault is so serious that the system function concerned fails to operate (Isermann 2006). The title failure detection has now been superseded by *fault detection*, e.g., in *fault detection and isolation* (FDI) or *fault detection and diagnosis* (FDD) (Chen and

Patton 1999; Gertler 1998; Patton et al. 2000) motivated by studies in the 1980s on this topic (Patton et al. 1989). *Fault-tolerant control* (FTC) began to develop in the early 1990s (Patton 1993) and is now a standard in the literature (Patton 1997; Blanke et al. 2006; Zhang and Jiang 2008), based on the aerospace subject of reconfigurable flight control making use of redundant actuators and sensors (Steinberg 2005; Edwards et al. 2010).

Definitions Relating to Fault-Tolerant Control

FTC is a strategy in control systems architecture and design to ensure that a closed-loop system can continue acceptable operation in the face of bounded actuator, sensor, or process faults. The goal of FTC design must ensure that the closed-loop system maintains satisfactory stability and acceptable performance during either one or more fault actions. When prescribed stability and closed-loop performance indices are maintained despite the action of faults, the system is said to be "fault tolerant," and the control scheme that ensures the fault tolerance is the fault-tolerant controller (Blanke et al. 2006; Patton 1997).

Fault modelling is concerned with the representation of the real physical faults and their effects on the system mathematical model. Fault modelling is important to establish how a fault should be detected, isolated, or compensated.

Fault-Tolerant Control, Fig. 1 Closed-loop system with actuator, process, and sensor faults
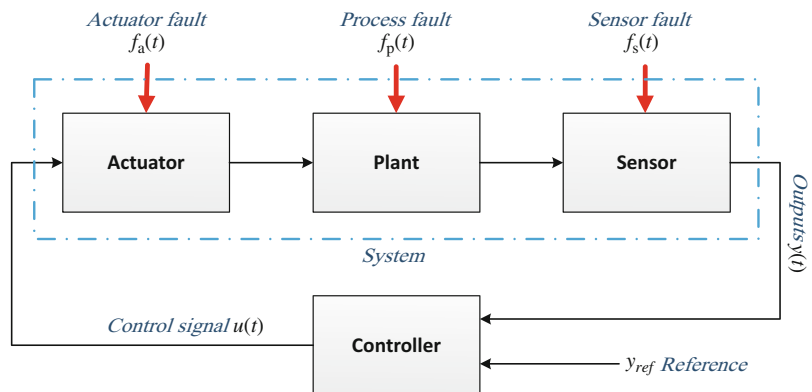
The faults illustrated in Fig. 1 act at system locations defined as follows (Chen and Patton 1999):

An ***actuator fault*** ($f_a(t)$) corresponds to variations of the control input $u(t)$ applied to the controlled system either completely or partially. The complete failure of an actuator means that it produces no actuation regardless of the input applied to it, e.g., as a result of breakage and burnout of wiring. For partial actuator faults, the actuator becomes less effective and provides the plant with only a part of the normal actuation signal.

A sensor is an item of equipment that takes a measurement or observation from the system, e.g., potentiometers, accelerometers, tachometers, pressure gauges, strain gauges, etc.; a ***sensor fault*** ($f_s(t)$) implies that incorrect measurements are taken from the real system. This fault can also be subdivided into either a complete or partial sensor fault. When a sensor fails, the measurements no longer correspond to the required physical parameters. For a partial sensor fault the measurements give an inaccurate indication of required physical parameters.

A ***process fault*** ($f_p(t)$) directly affects the physical system parameters and in turn the input/output properties of the system. Process faults are often termed *component faults*, arising as variations from the structure or parameters used during system modelling, and as such cover a wide class of possible faults, e.g., dirty water having a different heat transfer coefficient compared to when it is clean, or changes in the viscosity of a liquid or components slowly degrading over time through wear and tear, aging, or environmental effects.

Architectures and Classification of FTC Schemes

FTC methods are classified according to whether they are "passive" or "active," using fixed or reconfigurable control strategies (Eterno et al. 1985). Various architectures have been proposed for the implementation of FTC schemes, for example, the structure of reconfigurable control

based on generalized internal model control (GIMC) has been proposed by Zhou and Ren (2001) and other studies by Niemann and Stoustrup (2005). Figure 2 shows a suitable architecture to encompass active and passive FTC methods in which a distinction is made between "execution" and "supervision" levels. The essential differences and requirements between the passive FTC (PFTC) and active FTC (AFTC).

PFTC is based solely on the use of robust control in which potential faults are considered as if they are uncertain signals acting in the closed-loop system. This can be related to the concept of reliable control (Veillette et al. 1992). PFTC requires no online information from the fault diagnosis (FDI/FDD/FE) function about the occurrence or presence of faults and hence it is not by itself and adaptive system and does not involve controller reconfiguration (Patton 1993, 1997; Šiljak 1980). PFTC approach can be used if the time window during which the system remains stabilizable in the presence of a fault is short; see, for example, the problem of the double inverted pendulum (Weng et al. 2007) which is unstable during a loop failure.

AFTC has *two* conceptual steps to provide the system with fault-tolerant capability (Blanke et al. 2006; Patton 1997; Zhang and Jiang 2008; Edwards et al. 2009):

- Equip the system with a mechanism to make it able to detect and isolate (or even estimate) the fault promptly, identify a faulty component, and select the required remedial action in to maintain acceptable operation performance. With no fault a *baseline controller* attenuates disturbances and ensures good stability and closed-loop tracking performance (Patton 1997), and the diagnostic (FDI/FDD/FE) block recognizes that the closed-loop system is fault-free with no control law change required (supervision level).
- Make use of supervision level information and adapt or reconfigure/restructure the controller parameters so that the required remedial activity can be achieved (execution level).

Figure 3 gives a classification of PFTC and AFTC methods (Patton 1997).

Fault-Tolerant Control, Fig. 2 Scheme of FTC (Adapted from Blanke et al. (2006))
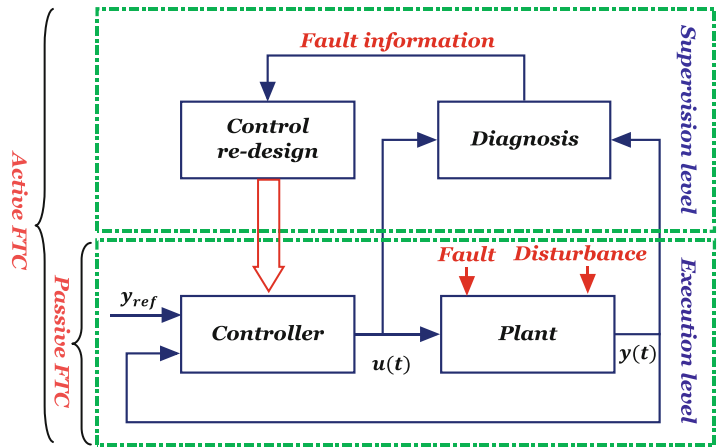
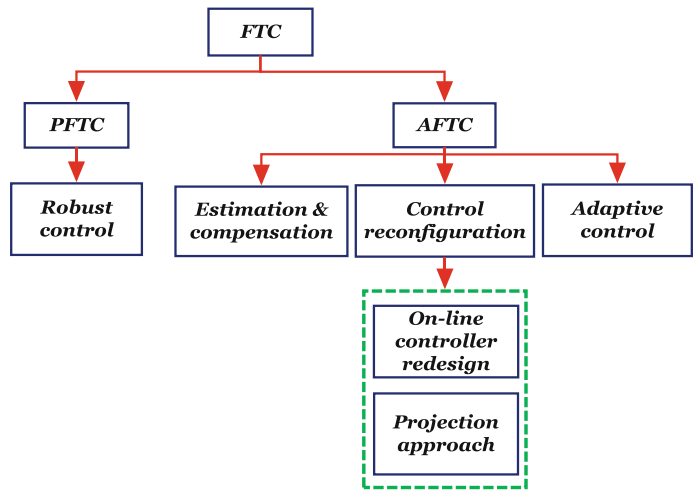**Fault-Tolerant Control, Fig. 3** General classification of FTC methods

F

Figure 3 shows that AFTC approaches are divided into two main types of methods: projection-based methods and online automatic controller redesign methods. The latter involves the calculation of new controller parameters following control impairment, i.e., using reconfigurable control. In projection-based methods, a new precomputed control law is selected according to the required controller structure (i.e., depending on the type of isolated fault).

AFTC methods use online-fault accommodation based on unanticipated faults, classified as (Patton 1997):

(a) Based on offline (pre-computed) control laws
(b) Online-fault accommodating

(c) Tolerant to unanticipated faults using FDI/FDD/FE
(d) Dependent upon use of a baseline controller

AFTC Examples

One example of AFTC is model-based predictive control (MPC) which uses online computed control redesign. MPC is online-fault accommodating; it does not use an FDI/FDD unit and is not dependent on a baseline controller. MPC has a certain degree of fault tolerance against actuator faults under some conditions even if the faults are not detected. The representation of actuator faults in MPC is relatively natural and straightforward since actuator faults such as jams and slew-rate reductions can be represented by changing the

MPC optimization problem constraints. Other faults can be represented by modifying the internal model used by MPC (Maciejowski 1998). The fact that online-fault information is not required means that MPC is an interesting method for flight control reconfiguration as demonstrated by Maciejowski and Jones in the GARTEUR AG 16 project on "fault-tolerant flight control" (Edwards et al. 2010).

Another interesting AFTC example that makes use of the concept of *model-matching* in explicit model following is the so-called pseudo-inverse method (PIM) (Gao and Antsaklis 1992) which requires the nominal or reference closed-loop system matrix to compute the new controller gain after a fault has occurred. The challenges are:

1. Guarantee of stability of the reconfigured closed-loop system
2. Minimization of the time consumed to approach the acceptable matching
3. Achieving perfect matching through use of different control methodologies

Exact model-matching may be too demanding, and some extensions to this approach make use of alternative, approximate (norm-based) model-matching through the computation of the required model-following gain. To relax the matching condition further, Staroswiecki (2005) proposed an *admissible model-matching* approach which was later extended by Tornil et al. (2010) using D-region pole assignment. The PIM approach requires an FDI/FDD/FE mechanism and is online-fault accommodating only in terms of *a priori* anticipated faults. This limits the practical value of this approach.

As a third example, feedback linearization can be used to compensate for nonlinear dynamic effects while also implementing control law reconfiguration or restructure. In flight control an aileron actuator fault will cause a strong coupling between the lateral and longitudinal aircraft dynamics. Feedback linearization is an established technique in flight control (Ochi and Kanai 1991). The faults are identified indirectly by estimating aircraft flight parameters online, e.g., using a recursive least-squares algorithm to update the FTC.

Hence, a AFTC system provides fault tolerance either by selecting a precomputed control law (*projection-based*) (Boskovic and Mehra 1999; Maybeck and Stevens 1991; Rauch 1995) or by synthesizing a new control strategy online (*online controller redesign*) (Ahmed-Zaid et al. 1991; Richter et al. 2007; Efimov et al. 2012; Zou and Kumar 2011).

Another widely studied AFTC method is the *estimation and compensation approach*, where a fault compensation input is superimposed on the nominal control input (Noura et al. 2000; Boskovic and Mehra 2002; Sami and Patton 2013; Zhang et al. 2004). There is a growing interest in robust FE methods based on sliding mode estimation (Edwards et al. 2000) and augmented observer methods (Gao and Ding 2007; Jiang et al. 2006; Sami and Patton 2013).

An important development of this approach is the so-called fault hiding strategy which is centered on achieving FTC loop goals such that the nominal control loop remains unchanged through the use of virtual actuators or virtual sensors (Blanke et al. 2006; Sami and Patton 2013). Fault hiding makes use of the difference between the nominal and faulty system state to changes in the system dynamics such that the required control objectives are continuously achieved even if a fault occurs. In the sensor fault case, the effect of the fault is hidden from the input of the controller. However, actuator faults are compensated by the effect of the fault (Lunze and Steffen 2006; Richter et al. 2007; Ponsart et al. 2010; Sami and Patton 2013) in which it is assumed that the FDI/FDD or FE scheme is available. The virtual actuator/sensor FTC can be good practical value if FDI/FDD/FE robustness can be demonstrated.

Traditional adaptive control methods that automatically adapt controller parameters to system changes can be used in a special application of AFTC, potentially removing the need for FDI/FDD and controller redesign steps (Tang et al. 2004; Zou and Kumar 2011) but possibly using the FE function. Adaptive control is suitable for FTC on plants that have slowly

varying parameters and can tolerate actuator and process faults. Sensor faults are not tolerated well as the controller parameters must adapt according to the faulty measurements, causing incorrect closed-loop system operation; the FDI/FDD/FE unit is required for such cases.

Summary and Future Directions

FTC is now a significant subject in control systems science with many quite significant application studies, particularly since the new millennium. Most of the applications are within the flight control field with studies such as the GARTEUR AG16 project "Fault-Tolerant Flight Control" (Edwards et al. 2010). As a very complex engineering-led and mathematically focused subject, it is important that FTC remains application-driven to keep the theoretical concepts moving in the right directions and satisfying end-user needs. The original requirement for FTC in safety-critical systems has now widened to encompass a good range of fault-tolerance requirements involving energy and economy, e.g., for greener aircraft and for FTC in renewable energy.

Faults and modelling uncertainties as well as endogenous disturbances have potentially competing effects on the control system performance and stability. This is the *robustness problem in FTC* which is beyond the scope of this article. The FTC system provides a degree of tolerance to closed-loop systems faults and it is also subject to the effects of modelling uncertainty arising from the reality that all engineering systems are nonlinear and can even have complex dynamics. For example, consider the PFTC approach relying on robustness principles – as a more complex extension to robust control. PFTC design requires the closed-loop system to be insensitive to faults as well as modelling uncertainties. This requires the use of multi-objective optimization methods e.g., using linear matrix inequalities (LMI), as well as methods of accounting for dynamical system parametric variations, e.g., linear parameter varying (LPV) system structures, Takagi-Sugeno, or sliding mode methods.

Cross-References

▶ Diagnosis of Discrete Event Systems
▶ Estimation, Survey on
▶ Fault Detection and Diagnosis
▶ H-infinity Control
▶ LMI Approach to Robust Control
▶ Lyapunov's Stability Theory
▶ Model Reference Adaptive Control
▶ Optimization Based Robust Control
▶ Robust Adaptive Control
▶ Robust Fault Diagnosis and Control
▶ Robust $\mathcal{H}_2$ Performance in Feedback Control

Bibliography

Ahmed-Zaid F, Ioannou P, Gousman K, Rooney R (1991) Accommodation of failures in the F-16 aircraft using adaptive control. IEEE Control Syst 11:73–78

Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2006) Diagnosis and fault-tolerant control. Springer, Berlin/New York

Boskovic JD, Mehra RK (1999) Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In: Proceedings of the ACC, San Diego, 2–4 June 1999, pp 1920–1924

Boskovic JD, Mehra RK (2002) An adaptive retrofit reconfigurable flight controller. In: Proceedings of the 41st IEEE CDC, Las Vegas, 10–13 Dec 2002, pp 1257–1262

Chen J, Patton RJ (1999) Robust model based fault diagnosis for dynamic systems. Kluwer, Boston

Edwards C, Spurgeon SK, Patton RJ (2000) Sliding mode observers for fault detection and isolation. Automatica 36:541–553

Edwards C, Lombaerts T, Smaili H (2010) Fault tolerant flight control a benchmark challenge. Springer, Berlin/Heidelberg

Efimov D, Cieslak J, Henry D (2012) Supervisory fault-tolerant control with mutual performance optimization. Int J Adapt Control Signal Process 27(4):251–279

Eterno J, Weiss J, Looze D, Willsky A (1985) Design issues for fault tolerant restructurable aircraft control. In: Proceedings of the 24th IEEE CDC, Fort-Lauderdale, Dec 1985

Gao Z, Antsaklis P (1992) Reconfigurable control system design via perfect model following. Int J Control 56:783–798

Gao Z, Ding S (2007) Actuator fault robust estimation and fault-tolerant control for a class of nonlinear descriptor systems. Automatica 43:912–920

Gertler J (1998) Fault detection and diagnosis in engineering systems. Marcel Dekker, New York

F

Isermann R (2006) Fault-diagnosis systems an introduction from fault detection to fault tolerance. Springer, Berlin/New York

Jiang BJ, Staroswiecki M, Cocquempot V (2006) Fault accommodation for nonlinear dynamic systems. IEEE Trans Autom Control 51:1578–1583

Lunze J, Steffen T (2006) Control reconfiguration after actuator failures using disturbance decoupling methods. IEEE Trans Autom Control 51:1590–1601

Maciejowski JM (1998) The implicit daisy-chaining property of constrained predictive control. J Appl Math Comput Sci 8(4):101–117

Maybeck P, Stevens R (1991) Reconfigurable flight control via multiple model adaptive control methods. IEEE Trans Aerosp Electron Syst 27:470–480

Niemann H, Stoustrup J (2005) An architecture for fault tolerant controllers. Int J Control 78(14):1091–1110

Noura H, Sauter D, Hamelin F, Theilliol D (2000) Fault-tolerant control in dynamic systems: application to a winding machine. IEEE Control Syst Mag 20:33–49

Ochi Y, Kanai K (1991) Design of restructurable flight control systems using feedback linearization. J Guid Dyn 14(5): 903–911

Patton RJ, Frank PM, Clark RN (1989) Fault diagnosis in dynamic Systems: theory and application. Prentice Hall, New York

Patton RJ (1993) Robustness issues in fault tolerant control. In: Plenary paper at international conference TOOLDIAG'93, Toulouse, Apr 1993

Patton RJ (1997) Fault tolerant control: the 1997 situation. In: IFAC Safeprocess '97, Hull, pp 1033–1055

Patton RJ, Frank PM, Clark RN (2000) Issues of fault diagnosis for dynamic systems. Springer, London/New York

Ponsart J, Theilliol D, Aubrun C (2010) Virtual sensors design for active fault tolerant control system applied to a winding machine. Control Eng Pract 18:1037–1044

Rauch H (1995) Autonomous control reconfiguration. IEEE Control Syst 15:37–48

Richter JH, Schlage T, Lunze J (2007) Control reconfiguration of a thermofluid process by means of a virtual actuator. IET Control Theory Appl 1:1606–1620

Sami M, Patton RJ (2013) Active fault tolerant control for nonlinear systems with simultaneous actuator and sensor faults. Int J Control Autom Syst 11(6):1149–1161

Šiljak DD (1980) Reliable control using multiple control systems. Int J Control 31:303–329

Staroswiecki M (2005) Fault tolerant control using an admissible model matching approach. In: Joint Decision and Control Conference and European Control Conference, Seville, 12–15 Dec 2005, pp 2421–2426

Steinberg M (2005) Historical overview of research in reconfigurable flight control. Proc IMechE, Part G J Aeros Eng 219:263–275

Tang X, Tao G, Joshi S (2004) Adaptive output feedback actuator failure compensation for a class of non-linear systems. Int J Adapt Control Signal Process 19(6): 419–444

Tornil S, Theilliol D, Ponsart JC (2010) Admissible model matching using $\mathcal{D}$R-regions: fault accommodation and robustness against FDD inaccuracies J Adapt Control Signal Process 24(11): 927–943

Veillette R, Medanic J, Perkins W (1992) Design of reliable control systems. IEEE Trans Autom Control 37:290–304

Weng J, Patton RJ, Cui P (2007) Active fault-tolerant control of a double inverted pendulum. J Syst Control Eng 221:895

Zhang Y, Jiang J (2008) Bibliographical review on reconfigurable fault-tolerant control systems. Annu Rev Control 32:229–252

Zhang X, Parisini T, Polycarpou M (2004) Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. IEEE Trans Autom Control 49(8):1259–1274

Zhang K, Jiang B, Staroswiecki M (2010) Dynamic output feedback-fault tolerant controller design for Takagi-Sugeno fuzzy systems with actuator faults. IEEE Trans Fuzzy Syst 18:194–201

Zhou K, Ren Z (2001) A new controller architecture for high performance, robust, and fault-tolerant control. IEEE Trans Autom Control 46(10):1613–1618

Zou A, Kumar K (2011) Adaptive fuzzy fault-tolerant attitude control of spacecraft. Control Eng Pract 19:10–21

FDD

▶ Fault Detection and Diagnosis

Feedback Linearization of Nonlinear Systems

A.J. Krener
Department of Applied Mathematics, Naval Postgrauate School, Monterey, CA, USA

Abstract

Effective methods exist for the control of linear systems but this is less true for nonlinear systems. Therefore, it is very useful if a nonlinear system can be transformed into or approximated by a linear system. Linearity is not invariant under nonlinear changes of state coordinates and nonlinear state feedback. Therefore, it may be possible to convert a nonlinear system into a linear one via these transformations. This is called feedback

linearization. This entry surveys feedback linearization and related topics.

Keywords

Distribution; Frobenius theorem; Involutive distribution; Lie derivative

Introduction

A controlled linear dynamics is of the form

$$\dot{x} = Fx + Gu \qquad (1)$$

where the state $x \in I\!\!R^n$ and the control $u \in I\!\!R^m$. A controlled nonlinear dynamics is of the form

$$\dot{x} = f(x, u) \qquad (2)$$

where x, u have the same dimensions but may be local coordinates on some manifolds $\mathcal{X}$, $\mathcal{U}$. Frequently, the dynamics is affine in the control, i.e.,

$$\dot{x} = f(x) + g(x)u \qquad (3)$$

where $f(x) \in I\!\!R^{n \times 1}$ is a vector field and $g(x) = [g^1(x), \dots, g^m(x)] \in I\!\!R^{n \times m}$ is a matrix field.

Linear dynamics are much easier to analyze and control than nonlinear dynamics. For example, to globally stabilize the linear dynamics (1), all we need to do is to find a linear feedback law $u = Kx$ such that all the eigenvalues of $F + GK$ are in the open left half plane. Finding a feedback law $u = \kappa(x)$ to globally stabilize the nonlinear dynamics is very difficult and frequently impossible. Therefore, finding techniques to linearize nonlinear dynamics has been a goal for several centuries.

The simplest example of a linearization technique is to approximate a nonlinear dynamics around a critical point by its first-order terms. Suppose x^0, u^0 is an operating point for the nonlinear dynamics (2), that is, $f(x^0, u^0) = 0$. Define displacement variables $z = x - x^0$ and $v = u - u^0$, and assuming $f(x, u)$ is smooth

around this operating point, expand (2) to first order

$$\dot{z} = \frac{\partial f}{\partial x}(x^0, u^0)z + \frac{\partial f}{\partial u}(x^0, u^0)v$$
$$+ O(z, v)^2 \qquad (4)$$

Ignoring the higher order terms, we get a linear dynamics (1) where

$$F = \frac{\partial f}{\partial x}(x^0, u^0), \quad G = \frac{\partial f}{\partial u}(x^0, u^0)$$

This simple technique works very well in many cases and is the basis for many engineering designs. For example, if the linear feedback $v = Kz$ puts all the eigenvalues of $F + GK$ in the left half plane, then the affine feedback $u = u^0 + K(x - x^0)$ makes the closed-loop dynamics locally asymptotically stable around x^0. So, one way to linearize a nonlinear dynamics is to approximate it by a linear dynamics.

The other way to linearize is by a nonlinear change of state coordinates and a nonlinear state feedback because linearity is not invariant under these transformations. To see this, suppose we have a controlled linear dynamics (1) and we make the nonlinear change of state coordinates $z = \phi(x)$ and nonlinear feedback $u = \gamma(z, v)$. We assume that these transformations are invertible from some neighborhood of $x^0 = 0, u^0 = 0$ to some neighborhood of z^0, v^0, and the inverse maps are

$$\psi(\phi(x)) = x, \quad \phi(\psi(z) = z$$
$$\kappa(\psi(z), \gamma(z, v)) = v, \quad \gamma(\phi(x), \kappa(x, u)) = u$$

then (1) becomes

$$\dot{z} = \frac{\partial \phi}{\partial x}(x)(Fx + Gu)$$
$$= \frac{\partial \phi}{\partial x}(\psi(z))(F\psi(z) + G\gamma(z, v))$$

which is a controlled nonlinear dynamics (2). This raises the question asked by Brockett (1978), when is a controlled nonlinear dynamics a change of coordinate and feedback away from a controlled linear dynamics?

F

Linearization of a Smooth Vector Field

Let us start by addressing an apparently simpler question that was first considered by Poincaré. Given an uncontrolled nonlinear dynamics around a critical point, say $x^0 = 0$,

$$\dot{x} = f(x), \qquad f(0) = 0,$$

find a smooth local change of coordinates

$$z = \phi(x), \quad \phi(0) = 0$$

which transforms it into an uncontrolled linear dynamics.

$$\dot{z} = Fz.$$

This question is apparently simpler, but as we shall see in the next section, the corresponding question for a controlled nonlinear dynamics that is affine in the control is actually easier to answer.

Without loss of generality, we can restrict our attention to changes of coordinates which carry $x^0 = 0$ to $z^0 = 0$ and whose Jacobian at this point is the identity, i.e.,

$$z = x + O(x^2)$$

then

$$F = \frac{\partial f}{\partial x}(0).$$

Poincaré's formal solution to this problem was to expand the vector field and the desired change of coordinates in a power series,

$$\dot{x} = Fx + f^{[2]}(x) + O(x^3)$$
$$z = x - \phi^{[2]}(x)$$

where $f^{[2]}$, $\phi^{[2]}$ are n-dimensional vector fields, whose entries are homogeneous polynomials of degree 2 in x. A straightforward calculation yields

$$\dot{z} = Fz + f^{[2]}(x) - \left[Fx, \phi^{[2]}(x)\right] + O(x)^3$$

where the Lie bracket of two vector fields $f(x)$, $g(x)$ is the new vector field defined by

$$[f(x), g(x)] = \frac{\partial g}{\partial x}(x)f(x) - \frac{\partial f}{\partial x}(x)g(x).$$

Hence, $\phi^{[2]}(x)$ must satisfy the so-called homological equation (Arnol'd 1983)

$$\left[Fx, \phi^{[2]}(x)\right] = f^{[2]}(x).$$

This is a linear equation from the space of quadratic vector fields to the space of quadratic vector fields. The quadratic n-dimensional vector fields form a vector space of dimension n times $n + 1$ choose 2.

Poincaré showed that the eigenvalues of the linear map

$$\phi^{[2]}(x) \mapsto \left[Fx, \phi^{[2]}(x)\right] \tag{5}$$

are $\lambda_i + \lambda_j - \lambda_k$ where $\lambda_i, \lambda_j, \lambda_k$ are eigenvalues of F. If none of these expressions are zero, then the operator (5) is invertible. A degree two resonance occurs when $\lambda_i + \lambda_j - \lambda_k = 0$ and then the homological equation is not solvable for all $f^{[2]}(x)$.

Suppose a change of coordinates exists that linearizes the vector field up to degree r. In the new coordinates, the vector field is of the form

$$\dot{x} = Fx + f^{[r]}(x) + O(x)^{r+1}.$$

We seek a change of coordinates of the form

$$z = x - \phi^{[r]}(x)$$

to cancel the degree r terms, i.e., we seek a solution of the rth degree homological equation,

$$\left[Fx, \phi^{[r]}(x)\right] = f^{[r]}(x).$$

A degree r resonance occurs if

$$\lambda_{i_1} + \ldots + \lambda_{i_r} - \lambda_k = 0.$$

If there is no resonance of degree r, then the degree r homological equation is uniquely solvable for every $f^{[r]}(x)$.

When there are no resonances of any degree, then the convergence of the formal power series solution is delicate. We refer the reader to Arnol'd (1983) for the details.

Linearization of a Controlled Dynamics by Change of State Coordinates

Given a controlled affine dynamics (3) when does there exist a smooth local change of coordinates

$$z = \phi(x), \qquad 0 = \phi(0)$$

transforming it to

$$\dot{z} = Fz + Gu$$

where

$$F = \frac{\partial f}{\partial x}(0), \qquad G = g(0)$$

This is an easier question to answer than that of Poincaré.

The controlled affine dynamics (3) is said to have well-defined controllability (Kronecker) indices if there exists a reordering of $g^1(x), \ldots, g^m(x)$ and integers $r_1 \geq r_2 \geq \cdots \geq r_m \geq 0$ such that $r_1 + \cdots + r_m = n$, and the vector fields

$$\{ad^k(f)g^j : j = 1, \ldots, m, \ k = 0, \ldots, r_j - 1\}$$

are linearly independent at each x where g^i denotes the ith column of g and

$$ad^0(f)g^i = g^i, \ \ ad^k(f)g^i = \left[f, ad^{k-1}(f)g^i\right].$$

If there are several sets of indices that satisfy this definition, then the controllability indices are the smallest in the lexicographic ordering.

A necessary and sufficient condition is that

$$\left[ad^k(f)g^i, ad^l(f)g^j\right] = 0$$

for $k = 0, \ldots, n-1, \ l = 0, \ldots, n$

The proof of this theorem is straightforward. Under a change of state coordinates, the vector fields and their Lie brackets are transformed by the Jacobian of the coordinate change. Trivially for linear systems,

$$ad^k(Fx)G^i = (-1)^k F^k G$$
$$\left[ad^k(Fx)G^i, ad^l(Fx)G^j\right] = 0.$$

Feedback Linearization

We turn to a question posed and partially answered by Brockett (1978). Given a system affine in the m-dimensional control

$$\dot{x} = f(x) + g(x)u,$$

find a smooth local change of coordinates and smooth feedback

$$z = \phi(x), \quad u = \alpha(x) + \beta(x)v$$

transforming it to

$$\dot{z} = Fz + Gv$$

Brockett solved this problem under the assumptions that β is a constant and the control is a scalar, $m = 1$. The more general question for $\beta(x)$ and arbitrary m was solved in different ways by Korobov (1979), Jakubczyk and Respondek (1980), Sommer (1980), Hunt and Su (1981), Su (1982), and Hunt et al. (1983).

We describe the solution when $m = 1$. If the pair F, G is controllable, then there exist an H such that

$$HF^{k-1}G = 0 \qquad k = 1, \ldots, n-1$$
$$HF^{n-1}G = 1$$

If the nonlinear system is feedback linearizable, then there exists a function $h(x) = H\phi(x)$ such that

$$L_{ad^{k-1}(f)g}h = 0 \qquad k = 1, \ldots, n-1$$

$$L_{ad^{n-1}(f)g}h \neq 0$$

where the Lie derivative of a function h by a vector field g is given by

$$L_g h = \frac{\partial h}{\partial x}g$$

This is a system of first-order PDEs, and the solvability conditions are given by the classical Frobenius theorem, namely, that

$$\{g, \ldots, ad^{n-2}(f)g\}$$

is involutive, i.e., its span is closed under Lie bracket.

For controllable systems, this is a necessary and sufficient condition. The controllability condition is that $\{g, \ldots, ad^{n-1}(f)g\}$ spans x space.

Suppose $m = 2$ and the system has controllability (Kronecker) indices $r_1 \geq r_2$. Such a system is feedback linearizable iff

$$\{g^1, , g^2, \ldots, ad^{r_i-2}(f)g^1, ad^{r_i-2}(f)g^2\}$$

is involutive for $i = 1, 2$. Another way of putting is that the distribution spanned by the first through rth rows of the following matrix must be involutive for $r = r_i - 1$, $i = 1, 2$. This is equivalent to the distribution spanned by the first through rth rows of the following matrix being involutive for all $r = 1, \ldots, r_1$.

$$\begin{bmatrix} g^1 & g^2 \\ ad(f)g & ad(f)g^2 \\ \vdots & \vdots \\ ad^{r_2-2}(f)g^1 & ad^{r_2-2}(f)g^2 \\ ad^{r_2-1}(f)g^1 & ad^{r_2-1}(f)g^2 \\ \vdots & \\ ad^{r_1-2}(f)g^1 & \\ ad^{r_1-1}(f)g^1 & \end{bmatrix}$$

One might ask if it is possible to use dynamic feedback to linearize a system that is not linearizable by static feedback. Suppose we treat one of the controls u_j as a state and let its derivative be a new control,

$$\dot{u}_j = \bar{u}_j$$

can the resulting system be linearized by state feedback and change of state coordinates? Loosely speaking, the effect of adding such an integrator to the jth control is to shift the jth column of the above matrix down by one row. This changes the distribution spanned by the first through rth rows of the above matrix and might make it involutive. A scalar input system $m = 1$ that is linearizable by dynamic state feedback is also linearizable by static state feedback. There are multi-input systems $m > 1$ that are dynamically linearizable but not statically linearizable (Charlet et al. 1989, 1991).

The generic system is not feedback linearizable, but mechanical systems with one actuator for each degree of freedom typically are feedback linearizable. This fact had been used in many applications, e.g., robotics, before the concept of feedback linearization.

One should not lose sight of the fact that stabilization, model-matching, or some other performance criterion is typically the goal of controller design. Linearization is a means to the goal. We linearize because we know how to meet the performance goal for linear systems.

Even when the system is linearizable, finding the linearizing coordinates and feedback can be a nontrivial task. Mechanical systems are the exception as the linearizing coordinates are usually the generalized positions. Since the $ad^{k-1}(f)g$ for $k = 1, \ldots, n-1$ are characteristic directions of the PDE for h, the general solutions of the ODE's

$$\dot{x} = ad^{k-1}(f)g(x)$$

can be used to construct the solution (Blankenship and Quadrat 1984). The Gardner-Shadwick (GS) algorithm (1992) is the most efficient method that is known.

Linearization of discrete time systems was treated by Lee et al. (1986). Linearization of discrete time systems around an equilibrium

manifold was treated by Barbot et al. (1995) and Jakubczyk (1987). Banaszuk and Hauser have also considered the feedback linearization of the transverse dynamics along a periodic orbit, (Banaszuk and Hauser 1995a,b).

Input–Output Linearization

Feedback linearization as presented above ignores the output of the system but typically one uses the input to control the output. Therefore, one wants to linearize the input–output response of the system rather than the dynamics. This was first treated in Isidori and Krener (1982) and Isidori and Ruberti (1984).

Consider a scalar input, scalar output system of the form

$$\dot{x} = f(x) + g(x)u, \qquad y = h(x)$$

The relative degree of the system is the number of integrators between the input and the output. To be more precise, the system is of relative degree $r \geq 1$ if for all x of interest,

$$L_{ad^j(f)g}h(x) = 0 \qquad j = 0, \ldots, r-2$$
$$L_{ad^{r-1}(f)g}h(x) \neq 0$$

In other words, the control appears first in the rth time derivative of the output. Of course, a system might not have a well-defined relative degree as the r might vary with x.

Rephrasing the result of the previous section, a scalar input nonlinear system is feedback linearizable if there exist an pseudo-output map $h(x)$ such the resulting scalar input, scalar output system has a relative degree equal to the state dimension n.

Assume we have a scalar input, scalar output system with a well-defined relative degree $1 \leq r \leq n$. We can define r partial coordinate functions

$$\zeta_i(x) = (L_f)^{i-1}h(x) \qquad i = 1, \ldots, r$$

and choose $n - r$ functions $\xi_i(x), i = 1, \ldots, n-r$ so that (ζ, ξ) are a full set of coordinates on the state space. Furthermore, it is always possible (Isidori 1995) to choose $\xi_i(x)$ so that

$$L_g\xi_i(x) = 0 \qquad i = 1, \ldots, n-r$$

In these coordinates, the system is in the normal form

$$y = \zeta_1$$
$$\dot{\zeta}_1 = \zeta_2$$
$$\vdots$$
$$\dot{\zeta}_{r-1} = \zeta_r$$
$$\dot{\zeta}_r = f_r(\zeta, \xi) + g_r(\zeta, \xi)u$$
$$\dot{\xi} = \phi(\zeta, \xi)$$

The feedback $u = u(\zeta, \xi, v)$ defined by

$$u = \frac{(v - f_r(\zeta, \xi))}{g_r(\zeta, \xi)}$$

transforms the system to

$$y = H\zeta$$
$$\dot{\zeta} = F\zeta + Gv$$
$$\dot{\xi} = \phi(\zeta, \xi)$$

where F, G, H are the $r \times r$, $r \times 1$, $1 \times r$ matrices

$$F = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \end{bmatrix}.$$

The system has been transformed into a string of integrators plus additional dynamics that is unobservable from the output.

By suitable choice of additional feedback $v = K\zeta$, one can insure that the poles of $F + GK$ are stable. The stability of the overall system then depends on the stability of the zero dynamics (Byrnes and Isidori 1984, 1988),

$$\dot{\xi} = \phi(0, \xi)$$

If this is stable, then the overall system will be stable. The zero dynamics is so-called because it is the dynamics that results from imposing the constraint $y(t) = 0$ on the system. For this to be satisfied, the initial value must satisfy $\zeta(0) = 0$ and the control must satisfy

$$u(t) = -\frac{f_r(0, \xi(t))}{g_r(0, \xi(t))}$$

Similar results hold in the multiple input, multiple output case, see Isidori (1995) for the details.

Approximate Feedback Linearization

Since so few controlled dynamics are exactly feedback linearizable, Krener (1984) introduced the concept of approximate feedback linearization. The goal is to find a smooth local change of coordinates and a smooth feedback

$$z = \phi(x), \qquad u = \alpha(x) + \beta(x)v$$

transforming the affinity controlled dynamics (3) to

$$\dot{z} = Fz + Gv + N(x, u)$$

where the nonlinearity $N(x, u)$ is small in some sense. In the design process, the nonlinearity is ignored, and the controller design is done on the linear model and then transformed back into a controller for the original system.

The power series approach of Poincaré was taken by Krener et al. (1987, 1988, 1991), Krener (1990), and Krener and Maag (1991). See also Kang (1994). It is applicable to dynamics which may not be affine in the control. The controlled nonlinear dynamics (2), the change of coordinates, and the feedback are expanded in a power series

$$\dot{x} = Fx + Gu + f^{[2]}(x, u) + O(x, u)^3$$

$$z = x - \phi^{[2]}(x)$$

$$v = u - \alpha^{[2]}(x, u)$$

The transformed system is

$$\dot{z} = Fz + Gv + f^{[2]}(x, u)$$

$$- \left[Fx + Gu, \phi^{[2]}(x) \right] + G\alpha^{[2]}(x, u)$$

$$+ O(x, u)^3$$

To eliminate the quadratic terms, one seeks a solution of the degree two homological equations for $\phi^{[2]}$, $\alpha^{[2]}$

$$\left[Fx + Gu, \phi^{[2]}(x) \right] - G\alpha^{[2]}(x, u) = f^{[2]}(x, u)$$

Unlike before, the degree two homological equations are not square. Almost always, the number of unknowns is less than the number of equations. Furthermore, the the mapping

$$\left(\phi^{[2]}(x), \alpha^{[2]}(x, u) \right) \mapsto \left[Fx + Gu, \phi^{[2]}(x) \right]$$

$$-G\alpha^{[2]}(x, u)$$

is less than full rank. Hence, only an approximate, e.g., a least squares solution, is possible. Krener has written a MATLAB toolbox (http://www.math.ucdavis.edu/ krener 1995) to compute term by term solutions to the homological equations. The routine **fh2f_h_.m** sequentially computes the least square solutions of the homological equations to arbitrary degree.

Observers with Linearizable Error Dynamics

The dual of linear state feedback is linear input–output injection. Linear input–output injection is the transformation carrying

$$\dot{x} = Fx + Gu$$

$$y = Hx$$

into

$$\dot{x} = Fx + Bu + Ly + Mu$$

$$y = Hx$$

Linear input–output injection and linear change of state coordinates

$$\dot{x} = Fx + Bu + Ly + Mu$$

$$y = Hx$$

$$z = Tx$$

$$\dot{z} = TFT^{-1}x + TLy + TMu$$

define a group action on the class of linear systems. Of course, output injection is not physically realizable on the original system, but it is realizable on the observer error dynamics.

Nonlinear input–output injection is not well defined independent of the coordinates; input–output injection in one coordinate system does not look like input–output injection in another coordinate system.

If a system

$$\dot{x} = f(x, u), \qquad y = h(x)$$

can be transformed by nonlinear changes of state and output coordinates

$$z = \phi(x), \qquad w = \gamma(y)$$

to a linear system with nonlinear input–output injection

$$\dot{z} = Fz + Gu + \alpha(y, u)$$

$$w = Hz$$

then the observer

$$\dot{\hat{z}} = (F + LH)\hat{z} + Gu + \alpha(y, u) - Lw$$

has linear error dynamics

$$\tilde{z} = z - \hat{z}$$

$$\dot{\tilde{z}} = (F + LH)\tilde{z}$$

If H, F is detectable, then $F + LH$ can be made Hurwitz, i.e., all its eigenvalues are in the open left half plane.

The case when $\gamma = $ identity, there are no inputs $m = 0$ and one output $p = 1$,

$$\dot{x} = f(x)$$

$$y = h(x)$$

was solved by Krener and Isidori (1983) and Bestle and Zeitz (1983) when the pair H, F defined by

$$F = \frac{\partial f}{\partial x}(0)$$

$$H = \frac{\partial h}{\partial x}(0)$$

is observable.

One seeks a change of coordinates $z = \phi(x)$ so that the system is linear up to output injection

$$\dot{z} = Fz + \alpha(y)$$

$$y = Hz$$

If they exist, the z coordinates satisfy the PDE's

$$L_{ad^{n-k}(f)g}(z_j) = \delta_{k,j}$$

where the vector field $g(x)$ is defined by

$$L_g L_f^{k-1} h = \begin{cases} 0 & 1 \le k < n \\ 1 & k = n \end{cases}$$

The solvability conditions for these PDE's are that for $1 \le k < l \le n - 1$

$$\left[ad^{k-1}(f)g, ad^{l-1}(f)g \right] = 0.$$

The general case with γ, m, p arbitrary was solved by Krener and Respondek (1985). The solution is a three-step process. First, one must set up and solve a linear PDE for $\gamma(y)$. The integrability conditions for this PDE involve the vanishing of a pseudo-curvature (Krener 1986). The next two steps are similar to the above.

One defines a vector field g^j, $1 \leq j \leq p$ for each output, these define a PDE for the change of coordinates, for which certain integrability conditions must be satisfied. The process is more complicated than feedback linearization and even less likely to be successful so approximate solutions must be sought which we will discuss later in this section. We refer the reader Krener and Respondek (1985) and related work Zeitz (1987) and Xia and Gao (1988a,b, 1989).

Very few systems can be linearized by change of state coordinates and input–output injection, so Krener et al. (1987, 1988, 1991), Krener (1990), and Krener and Maag (1991) sought approximate solutions by the power series approach. Again, the system, the changes of coordinates, and the output injection are expanded in a power series. See the above references for details.

Conclusion

We have surveyed the various ways a nonlinear system can be approximated by a linear system.

Cross-References

- ▶ Differential Geometric Methods in Nonlinear Control
- ▶ Lie Algebraic Methods in Nonlinear Control
- ▶ Nonlinear Zero Dynamics

Bibliography

Arnol'd VI (1983) Geometrical methods in the theory of ordinary differential equations. Springer, Berlin

Banaszuk A, Hauser J (1995a) Feedback linearization of transverse dynamics for periodic orbits. Syst Control Lett 26:185–193

Banaszuk A, Hauser J (1995b) Feedback linearization of transverse dynamics for periodic orbits in $\mathbf{R}^3$ with points of transverse controllability loss. Syst Control Lett 26:95–105

Barbot JP, Monaco S, Normand-Cyrot D (1995) Linearization about an equilibrium manifold in discrete time. In: Proceedings of IFAC NOLCOS 95, Tahoe City. Pergamon

Barbot JP, Monaco S, Normand-Cyrot D (1997) Quadratic forms and approximate feedback linearization in discrete time. Int J Control 67:567–586

Bestle D, Zeitz M (1983) Canonical form observer design for non-linear time-variable systems. Int J Control 38:419–431

Blankenship GL, Quadrat JP (1984) An expert system for stochastic control and signal processing. In: Proceedings of IEEE CDC, Las Vegas. IEEE, pp 716–723

Brockett RW (1978) Feedback invariants for nonlinear systems. In: Proceedings of the international congress of mathematicians, Helsinki, pp 1357–1368

Brockett RW (1981) Control theory and singular Riemannian geometry. In: Hilton P, Young G (eds) New directions in applied mathematics. Springer, New York, pp 11–27

Brockett RW (1983) Nonlinear control theory and differential geometry. In: Proceedings of the international congress of mathematicians, Warsaw, pp 1357–1368

Brockett RW (1996) Characteristic phenomena and model problems in nonlinear control. In: Proceedings of the IFAC congress, Sidney. IFAC

Byrnes CI, Isidori A (1984) A frequency domain philosophy for nonlinear systems. In: Proceedings, IEEE conference on decision and control, Las Vegas. IEEE, pp 1569–1573

Byrnes CI, Isidori A (1988) Local stabilization of minimum-phase nonlinear systems. Syst Control Lett 11:9–17

Charlet R, Levine J, Marino R (1989) On dynamic feedback linearization. Syst Control Lett 13:143–151

Charlet R, Levine J, Marino R (1991) Sufficient conditions for dynamic state feedback linearization. SIAM J Control Optim 29:38–57

Gardner RB, Shadwick WF (1992) The GS-algorithm for exact linearization to Brunovsky normal form. IEEE Trans Autom Control 37:224–230

Hunt LR, Su R (1981) Linear equivalents of nonlinear time varying systems. In: Proceedings of the symposium on the mathematical theory of networks and systems, Santa Monica, pp 119–123

Hunt LR, Su R, Meyer G (1983) Design for multi-input nonlinear systems. In: Millman RS, Brockett RW, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston, pp 268–298

Isidori A (1995) Nonlinear control systems. Springer, Berlin

Isidori A, Krener AJ (1982) On the feedback equivalence of nonlinear systems. Syst Control Lett 2:118–121

Isidori A, Ruberti A (1984) On the synthesis of linear input–output responses for nonlinear systems. Syst Control Lett 4:17–22

Jakubczyk B (1987) Feedback linearization of discrete-time systems. Syst Control Lett 9:17–22

Jakubczyk B, Respondek W (1980) On linearization of control systems. Bull Acad Polonaise Sci Ser Sci Math 28:517–522

Kang W (1994) Approximate linearization of nonlinear control systems. Syst Control Lett 23: 43–52

Korobov VI (1979) A general approach to the solution of the problem of synthesizing bounded controls in a control problem. Math USSR-Sb 37:535

Krener AJ (1973) On the equivalence of control systems and the linearization of nonlinear systems. SIAM J Control 11:670–676

Krener AJ (1984) Approximate linearization by state feedback and coordinate change. Syst Control Lett 5:181–185

Krener AJ (1986) The intrinsic geometry of dynamic observations. In: Fliess M, Hazewinkel M (eds) Algebraic and geometric methods in nonlinear control theory. Reidel, Amsterdam, pp 77–87

Krener AJ (1990) Nonlinear controller design via approximate normal forms. In: Helton JW, Grunbaum A, Khargonekar P (eds) Signal processing, Part II: control theory and its applications. Springer, New York, pp 139–154

Krener AJ, Isidori A (1983) Linearization by output injection and nonlinear observers. Syst Control Lett 3: 47–52

Krener AJ, Maag B (1991) Controller and observer design for cubic systems. In: Gombani A, DiMasi GB, Kurzhansky AB (eds) Modeling, estimation and control of systems with uncertainty. Birkhauser, Boston, pp 224–239

Krener AJ, Respondek W (1985) Nonlinear observers with linearizable error dynamics. SIAM J Control Optim 23:197–216

Krener AJ, Karahan S, Hubbard M, Frezza R (1987) Higher order linear approximations to nonlinear control systems. In: Proceedings, IEEE conference on decision and control, Los Angeles. IEEE, pp 519–523

Krener AJ, Karahan S, Hubbard M (1988) Approximate normal forms of nonlinear systems. In: Proceedings, IEEE conference on decision and control, San Antonio. IEEE, pp 1223–1229

Krener AJ, Hubbard M, Karahan S, Phelps A, Maag B (1991) Poincare's linearization method applied to the design of nonlinear compensators. In: Jacob G, Lamnahbi-Lagarrigue F (eds) Algebraic computing in control. Springer, Berlin, pp 76–114

Lee HG, Arapostathis A, Marcus SI (1986) Linearization of discrete-time systems. Int J Control 45:1803–1822

Murray RM (1995) Nonlinear control of mechanical systems: a Lagrangian perspective. In: Krener AJ, Mayne DQ (eds) Nonlinear control systems design. Pergamon, Oxford

Nonlinear Systems Toolbox available for down load at http://www.math.ucdavis.edu/~krener/1995

Sommer R (1980) Control design for multivariable nonlinear time varying systems. Int J Control 31:883–891

Su R (1982) On the linear equivalents of control systems. Syst Control Lett 2:48–52

Xia XH, Gao WB (1988a) Nonlinear observer design by canonical form. Int J Control 47: 1081–1100

Xia XH, Gao WB (1988b) On exponential observers for nonlinear systems. Syst Control Lett 11: 319–325

Xia XH, Gao WB (1989) Nonlinear observer design by observer error linearization. SIAM J Control Optim 27:199–216

Zeitz M (1987) The extended Luenberger observer for nonlinear systems. Syst Control Lett 9: 149–156

Feedback Stabilization of Nonlinear Systems

A. Astolfi
Department of Electrical and Electronic Engineering, Imperial College London, London, UK
Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

Abstract

We consider the simplest design problem for nonlinear systems: the problem of rendering asymptotically stable a given equilibrium by means of state feedback. For such a problem, we provide a necessary condition, known as Brockett condition, and a sufficient condition, which relies upon the definition of a class of functions, known as control Lyapunov functions. The theory is illustrated by means of a few examples. In addition, we discuss a nonlinear enhancement of the so-called separation principle for stabilization by means of partial state information.

Keywords

Brockett theorem; Control Lyapunov function; Output feedback; State feedback

Introduction

The problem of feedback stabilization, namely, the problem of designing a feedback control law locally, or globally, asymptotically stabilizing a given equilibrium point, is the simplest design

problem for nonlinear systems. If the state of the system is available for feedback, then the problem is referred to as the state feedback stabilization problem, whereas if only part of the state, for example, an output signal, is available for feedback, the problem is referred to as the partial state feedback (or output feedback) stabilization problem. We initially focus on the state feedback stabilization problem, which can be formulated as follows.

Consider a nonlinear system described by the equation

$$\dot{x} = F(x, u), \tag{1}$$

where $x(t) \in IR^n$ denotes the state of the system, $u(t) \in IR^m$ denotes the input of the system, and $F : IR^n \times IR^m \to IR^n$ is a smooth mapping.

Let $x_0 \in IR^n$ be an *achievable* equilibrium, i.e., x_0 is such that there exists a constant $u_0 \in IR^m$ such that $F(x_0, u_0) = 0$. The state feedback stabilization problem consists in finding, if possible, a state feedback control law, described by the equation

$$u = \alpha(x), \tag{2}$$

with $\alpha : IR^n \to IR^m$, such that the equilibrium x_0 is a locally asymptotically stable equilibrium for the closed-loop system

$$\dot{x} = F(x, \alpha(x)). \tag{3}$$

Alternatively, one could require that the equilibrium be globally asymptotically stable. Note that it is not always possible to *extend* local properties to global properties. For example, for the system described by the equations $\dot{x}_1 = x_2(1 - x_1^2)$, $\dot{x}_2 = u$, with $x_1(t) \in IR$, $x_2(t) \in IR$, and $u(t) \in IR$, it is not possible to design a feedback law which renders the zero equilibrium globally asymptotically stable.

If only partial information on the state is available, then one has to resort to a dynamic *output* feedback controller, namely, a controller described by equations of the form

$$\dot{\xi} = \beta(\xi, y), \qquad u = \alpha(\xi), \tag{4}$$

where $\xi(t) \in IR^\nu$ describes the state of the controller, $y(t) \in IR^p$ is given by $y = h(x)$, for some mapping $h : IR^n \to IR^p$, and describes the available information on the state x, and $\beta : IR^\nu \times IR^p \to IR^\nu$ and $\alpha : IR^\nu \to IR^m$ are smooth mappings. Within this scenario, the stabilization problem boils down to selecting the (nonnegative) integer ν (i.e., the order of the controller), a constant $\xi_0 \in IR^\nu$, and the mappings α and β such that the closed-loop system

$$\dot{x} = F(x, \alpha(\xi)), \qquad \dot{\xi} = \beta(\xi, h(x)), \tag{5}$$

has a locally (or globally) asymptotically stable equilibrium at (x_0, ξ_0). Alternatively, one may require that the equilibrium (x_0, ξ_0) of the closed-loop system (5) be locally asymptotically stable with a region of attraction that contains a given, user-specified, set.

The rest of the entry is organized as follows. We begin discussing two key results. The first is a necessary condition, due to R.W. Brockett, for continuous stabilizability. This provides an obstruction to the solvability of the problem and can be used to show that, for nonlinear systems, controllability does not imply stabilizability by continuous feedback. The second one is the extension of the Lyapunov direct method to systems with control. The main idea is the introduction of a control version of Lyapunov functions, the *control Lyapunov functions*, which can be used to design stabilizing control laws by means of a *universal formula*. We then describe two classes of systems for which it is possible to construct, with systematic procedures, smooth control laws yielding global asymptotic stability of a given equilibrium: systems in feedback and in feedforward form. There are several other constructive and systematic stabilization methods which have been developed in the last few decades. Worth mentioning are passivity-based methods and center manifold-based methods.

We conclude the entry describing a nonlinear version of the separation principle for the asymptotic stabilization, by output feedback, of a general class of nonlinear systems.

Preliminary Results

To highlight the difficulties and peculiarities of the nonlinear stabilization problem, we recall some basic facts from linear systems theory and exploit such facts to derive a sufficient condition and a necessary condition. In the case of linear systems, i.e., systems described by the equation $\dot{x} = Ax + Bu$, with $A \in IR^{n \times n}$ and $B \in IR^{n \times m}$, and linear state feedback, i.e., feedback described by the equation $u = Kx$, with $K \in IR^{m \times n}$, the stabilization problem boils down to the problem of placing, in the complex plane, the eigenvalues of the matrix $A + BK$ to the left of the imaginary axis. This problem is solvable if and only if the uncontrollable modes of the system are located, in the complex plane, to the left of the imaginary axis.

The linear theory may be used to provide a simple obstruction to feedback stabilizability and a simple sufficient condition. Let x_0 be an achievable equilibrium with $u_0 = 0$ and note that the linear approximation of the system (1) around x_0 is described by an equation of the form $\dot{x} = Ax + Bu$.

If for any $K \in IR^{m \times n}$ the condition

$$\sigma(A + BK) \cap C^+ \neq \emptyset \qquad (6)$$

holds, then the equilibrium of the nonlinear system cannot be stabilized by any continuously differentiable feedback such that $\alpha(x_0) = 0$. The notation $\sigma(A)$ denotes the spectrum of the matrix A, i.e., the eigenvalues of A. Note however that, if the condition $\alpha(x_0) = 0$ is dropped, the obstruction does not hold: the zero equilibrium of $\dot{x} = x + xu$ is not stabilizable by any continuous feedback such that $\alpha(0) = 0$, yet the *feedback* $u = -2$ is a (global) stabilizer.

On the contrary, if there exists a K such that

$$\sigma(A + BK) \subset C^-$$

then the feedback $\alpha(x) = Kx$ locally asymptotically stabilizes the equilibrium x_0 of the closed-loop system. This fact is often referred to as *the linearization approach*.

The above linear arguments are often inadequate to design feedback stabilizers: a theory for nonlinear feedback has to be developed. However, this theory is much more involved. In particular, it is important to observe that the solvability of the stabilization problem may depend upon the regularity properties of the feedback, i.e., of the mapping α. In fact, a given equilibrium of a nonlinear system may be rendered locally asymptotically stable by a continuous feedback, whereas there may be no continuously differentiable feedback achieving the same goal. If the feedback is required to be continuously differentiable, then the problem is often referred to as the *smooth* stabilization problem.

Example 1 To illustrate the role of the regularity properties of the feedback, consider the system described by the equations

$$\dot{x}_1 = x_1 - x_2^3, \qquad \dot{x}_2 = u,$$

with $x_1(t) \in IR$, $x_2(t) \in IR$, and $u(t) \in IR$, and the equilibrium $x_0 = (0,0)$. The equilibrium is globally asymptotically stabilized by the continuous feedback

$$\alpha(x) = -x_2 + x_1 + \frac{4}{3}x_1^{\frac{1}{3}} - x_2^3,$$

but it is not stabilizable by any continuously differentiable feedback. Note, in fact, that condition (6) holds.

Brockett Theorem

Brockett's necessary condition, which is far from being sufficient, provides a simple test to rule out the existence of a continuous stabilizer.

Theorem 1 *Consider the system (1) and assume $x_0 = 0$ is an achievable equilibrium with $u_0 = 0$.*

Assume there exists a continuous stabilizing feedback $u = \alpha(x)$. Then, for each $\epsilon > 0$ there exists $\delta > 0$ such that, for all y with $||y|| < \delta$, the equation $y = F(x, u)$ has at least one solution in the set $||x|| < \epsilon$, $||u|| < \epsilon$.

Theorem 1 can be reformulated as follows. The existence of a continuous stabilizer implies that the image of the mapping $F : IR^n \times IR^m \to IR^n$ covers a neighborhood of the origin. Note, in addition, that the obstruction expressed by Theorem 1 is of topological nature. Hence, it requires continuity of F and α, and time invariance: it does not hold if $u = \alpha(x, t)$, i.e., a time-varying feedback is designed.

In the linear case, Brockett condition reduces to the condition

$$\text{rank}[A - \lambda I, \ B] = n$$

for $\lambda = 0$. This is a necessary, but clearly not sufficient, condition for the stabilizability of $\dot{x} = Ax + Bu$.

Example 2 Consider the kinematic model of a mobile robot given by the equations

$$\begin{aligned} \dot{x} &= \cos \theta \, v, \\ \dot{y} &= \sin \theta \, v, \\ \dot{\theta} &= \omega, \end{aligned}$$

where $(x(t), y(t)) \in IR^2$ denotes the Cartesian position of the robot, $\theta(t) \in (-\pi, \pi]$ denotes the robot orientation (with respect to the x-axis), $v(t) \in IR$ is the forward velocity of the robot, and $\omega(t) \in IR$ is its angular velocity. Simple intuitive considerations suggest that the system is controllable, i.e., it is possible to select the forward and angular velocities to drive the robot from any initial position/orientation to any final position/orientation in any given positive time. Nevertheless, the zero equilibrium (and any other equilibrium of the system) is not continuously stabilizable. In fact, the equations

$$y_1 = \cos \theta \, v, \qquad y_2 = \sin \theta \, v, \qquad y_3 = \omega,$$

with $||(y_1, y_2, y_3)|| < \delta$ and $||(x, y, \theta)|| < \epsilon$, $||(v, \omega)|| < \epsilon$, are in general not solvable. For example, if $\epsilon < \pi/2$ and $y_1 = 0$, $y_2 \neq 0$, $y_3 = 0$, then the unique solution of the first and third equations is $v = 0$ and $\omega = 0$, implying $\sin \theta \, v = 0$; hence, the second equation does not have a solution.

Control Lyapunov Functions

The Lyapunov theory states that the equilibrium x_0 of the system

$$\dot{x} = f(x),$$

with $f : IR^n \to IR^n$, is locally asymptotically stable if there exists a continuously differentiable function $V : IR^n \to IR$, called Lyapunov function, and a neighborhood U of x_0 such that $V(x_0) = 0$, $V(x) > 0$, for all $x \in U$ and $x \neq x_0$, and $\frac{\partial V}{\partial x} f(x) < 0$, for all $x \in U$ and $x \neq x_0$.

To apply this idea to the stabilization problem, consider the system (1). If the equilibrium x_0 of $\dot{x} = F(x, u)$ is continuously stabilizable, then there must exist a continuously differentiable function V and a neighborhood U of x_0 such that

$$\inf_u \frac{\partial V}{\partial x} F(x, u) < 0,$$

for all $x \in U$ and $x \neq x_0$. This motivates the following definition.

Definition 1 A continuously differentiable function V such that

- $V(x_0) = 0$ and $V(x) > 0$, for all $x \in U$ and $x \neq x_0$,
- $\inf_u \frac{\partial V}{\partial x} F(x, u) < 0$, for all $x \in U$ and $x \neq x_0$,

is called a *control Lyapunov function*.

By Lyapunov theory, the existence of a continuous stabilizer implies the existence of a control Lyapunov function. On the other hand, the existence of a control Lyapunov function does not guarantee the existence of a stabilizer. However, in the case of systems affine in the control, i.e., systems described by the equation

$$\dot{x} = f(x) + g(x)u, \tag{7}$$

with $f : IR^n \to IR^n$ and $g : IR^n \to IR^{n \times m}$ smooth mappings, very general results can be proven. These have been proven by Z. Artstein, who gave a nonconstructive statement, and have been given a constructive form by E.D. Sontag.

In particular, for single-input nonlinear systems, the following statement holds.

Theorem 2 *Consider the system (7), with $m = 1$, and assume $f(0) = 0$.*

There exists an almost smooth feedback, i.e., the feedback $\alpha(x)$ is continuously differentiable for all $x \in IR^n$ and $x \neq 0$, and continuous at $x = 0$ which globally asymptotically stabilizes the equilibrium $x = 0$ if and only if there exists a positive definite, radially unbounded, i.e., $\lim_{\|x\| \to \infty} V(x) = \infty$ and smooth function $V(x)$ such that

1. $\dfrac{\partial V}{\partial x} g(x) = 0 \quad \Rightarrow \quad \dfrac{\partial V}{\partial x} f(x) < 0$, *for all $x \neq 0$;*
2. *For each $\epsilon > 0$ there is a $\delta > 0$ such that $\|x\| < \delta$ implies that there is a $|u| < \epsilon$ such that*

$$\frac{\partial V}{\partial x} f(x) + \frac{\partial V}{\partial x} g(x) u < 0.$$

Condition 2 is known as the *small control property*, and it is necessary to guarantee continuity of the feedback at $x = 0$. If Conditions 1 and 2 hold, then an almost smooth feedback is given by the so-called Sontag's universal formula:

$$\alpha(x) = \begin{cases} 0, & \text{if } \dfrac{\partial V}{\partial x} g(x) = 0, \\[4mm] -\dfrac{\frac{\partial V}{\partial x} f(x) + \sqrt{\left(\frac{\partial V}{\partial x} f(x)\right)^2 + \left(\frac{\partial V}{\partial x} g(x)\right)^4}}{\frac{\partial V}{\partial x} g(x)}, & \text{elsewhere.} \end{cases}$$

Constructive Stabilization

We now introduce two classes of nonlinear systems for which systematic design methods to solve the state feedback stabilization problem are available.

Feedback Systems

Consider a nonlinear system described by equations of the form

$$\dot{x}_1 = f_1(x_1, x_2), \qquad \dot{x}_2 = u, \qquad (8)$$

with $x_1(t) \in IR^n$, $x_2(t) \in IR$, $u(t) \in IR^n$ and $f_1(0, 0) = 0$. This system belongs to the so-called class of feedback systems for which a sort of reduction principle holds: the zero equilibrium of the system is smoothly stabilizable if the same holds for the *reduced* system $\dot{x}_1 = f(x_1, v)$, which is obtained from the first of Eq. (8) replacing the state variable x_2 with a *virtual control* input v. To show this property, suppose there exist a continuously differentiable function $\alpha_1 : IR^n \to IR$ and

a continuously differentiable and radially unbounded function $V_1 : IR^n \to IR$ such that $V_1(0) = 0$, $V_1(x_1) > 0$, for all $x_1 \neq 0$, and

$$\frac{\partial V_1}{\partial x_1} f(x_1, \alpha_1(x_1)) < 0,$$

for all $x_1 \neq 0$, i.e., the zero equilibrium of the system $\dot{x}_1 = f(x_1, v)$ is globally asymptotically stabilizable.

Consider now the function

$$V(x_1, x_2) = V_1(x_1) + \frac{1}{2}(x_2 - \alpha_1(x_1))^2,$$

which is radially unbounded and such that $V(0, 0) = 0$ and $V(x_1, x_2) > 0$ for all nonzero (x_1, x_2), and note that

$$\begin{aligned} \dot{V} &= \frac{\partial V_1}{\partial x_1} f(x_1, x_2) + (x_2 - \alpha_1(x_1)) \\ &\quad \times (u + \Delta_1(x_1, x_2)) \\ &= \frac{\partial V_1}{\partial x_1} f(x_1, \alpha_1(x_1)) + (x_2 - \alpha_1(x_1)) \\ &\quad \times (u + \Delta_2(x_1, x_2)), \end{aligned}$$

for some continuously differentiable mappings Δ_1 and Δ_2. As a result, the feedback

$$\alpha(x_1, x_2) = -\Delta_2(x_1, x_2) - k(x_2 - \alpha_1(x_1)),$$

with $k > 0$, yields $\dot{V} < 0$ for all nonzero (x_1, x_2); hence, the feedback is a continuously differentiable stabilizer for the zero equilibrium of the system (8). Note, finally, that the function V is a control Lyapunov for the system (8); hence, Sontag's formula can be also used to construct a stabilizer.

The result discussed above is at the basis of the so-called backstepping technique for recursive stabilization of systems described, for example, by equations of the form

$$\begin{aligned}
\dot{x}_1 &= x_2 + \varphi_1(x_1), \\
\dot{x}_2 &= x_3 + \varphi_2(x_1, x_2), \\
\dot{x}_3 &= x_4 + \varphi_3(x_1, x_2, x_3), \\
&\vdots \\
\dot{x}_n &= u + \varphi_n(x_1, \ldots, x_n),
\end{aligned}$$

with $x_i(t) \in I\!R$ for all $i \in [1, n]$, and φ_i smooth mappings such that $\varphi_i(0) = 0$, for all $i \in [1, n]$.

Feedforward Systems
Consider a nonlinear system described by equations of the form

$$\dot{x}_1 = f_1(x_2), \qquad \dot{x}_2 = f_2(x_2) + g_2(x_2)u, \quad (9)$$

with $x_1(t) \in I\!R$, $x_2(t) \in I\!R^n$, $u(t) \in I\!R$, $f_1(0) = 0$ and $f_2(0) = 0$. This system belongs to the so-called class of feedforward systems for which, similarly to feedback systems, a sort of reduction principle holds: the zero equilibrium of the system is smoothly stabilizable if the zero equilibrium of the *reduced* system $\dot{x}_2 = f(x_2)$ is globally asymptotically stable and some additional structural assumption holds. To show this property, suppose there exists a continuously differentiable and radially unbounded function $V_2 : I\!R^n \to I\!R$ such that $V_2(0) = 0$, $V_2(x_2) > 0$, for all $x_2 \neq 0$, and

$$\frac{\partial V_2}{\partial x_2} f_2(x_2) < 0,$$

for all $x_2 \neq 0$. Suppose, in addition, that there exists a continuously differentiable mapping $M(x_2)$ such that

$$f_1(x_2) - \frac{\partial M}{\partial x_2} f_2(x_2) = 0,$$

$M(0) = 0$ and $\left. \dfrac{\partial M}{\partial x_2} \right|_{x_2=0} \neq 0$. Existence of such a mapping is guaranteed, for example, by asymptotic stability of the linearization of the system $\dot{x}_2 = f_2(x_2)$ around the origin and controllability of the linearization of the system (9) around the origin.

Consider now the function

$$V(x_1, x_2) = \frac{1}{2}(x_1 - M(x_2))^2 + V_2(x_2),$$

which is radially unbounded and such that $V(0, 0) = 0$ and $V(x_1, x_2) > 0$ for all nonzero (x_1, x_2), and note that

$$\begin{aligned}
\dot{V} &= -(x_1 - M(x_2))\frac{\partial M}{\partial x_2} g_2(x_2)u + \frac{\partial V_2}{\partial x_2} f_2(x_2) \\
&\quad + \frac{\partial V_2}{\partial x_2} g_2(x_2)u \\
&= \frac{\partial V_2}{\partial x_2} f_2(x_2) + \left(\frac{\partial V_2}{\partial x_2} - (x_1 - M(x_2))\frac{\partial M}{\partial x_2} \right) \\
&\quad \times g_2(x_2)u.
\end{aligned}$$

As a result, the feedback

$$\begin{aligned}
\alpha(x_1, x_2) = -k &\left(\frac{\partial V_2}{\partial x_2} - (x_1 - M(x_2))\frac{\partial M}{\partial x_2} \right) \\
&\times g_2(x_2),
\end{aligned}$$

with $k > 0$, yields $\dot{V} < 0$ for all nonzero (x_1, x_2); hence, the feedback is a continuously differentiable stabilizer for the zero equilibrium of the system (9). Note, finally, that the function V is a control Lyapunov for the system (9); hence, Sontag's formula can be also used to construct a stabilizer.

The result discussed above is at the basis of the so-called forwarding technique for recursive stabilization of systems described, for example, by equations of the form

$$\begin{aligned}
\dot{x}_1 &= \varphi_1(x_2, \ldots, x_n), \\
\dot{x}_2 &= \varphi_2(x_3, \ldots, x_n), \\
&\vdots \\
\dot{x}_{n-1} &= \varphi_{n-1}(x_n), \\
\dot{x}_n &= u,
\end{aligned}$$

with $x_i(t) \in I\!R$ for all $i \in [1, n]$, and φ_i smooth mappings such that $\varphi_i(0) = 0$, for all $i \in [1, n]$.

Stabilization via Output Feedback

In the previous sections, we have studied the stabilization problem for nonlinear systems under the assumption that the whole state is available for feedback. This requires the online measurement of the state vector x, which may pose a severe constrain in applications. This observation motivates the study of the much more challenging, but more realistic, problem of stabilization with partial state information. This problem requires the introduction of a notion of observability. Note that for nonlinear systems, it is possible to define several, nonequivalent, observability notions. Similarly to section "Control Lyapunov Functions", we focus on the class of systems affine in the control, i.e., systems described by equations of the form

$$\begin{aligned}
\dot{x} &= f(x) + g(x)u, \\
y &= h(x),
\end{aligned} \tag{10}$$

with $f : I\!R^n \to I\!R^n$, $g : I\!R^n \to I\!R^{n \times m}$ and $h : I\!R^n \to I\!R^p$ smooth mappings. This is precisely the class of systems in Eq. (7) with the addition of the *output map h*, i.e., a map which describes the information that is available for feedback. In addition, we assume, to simplify the notation, that $m = 1$ and $p = 1$: the system is single input, single output. Finally assume, without loss of generality, that the equilibrium to be stabilized is $x_0 = 0$, that any stabilizing state feedback control law $u = \alpha(x)$ is such that $\alpha(0) = 0$, and that $h(0) = 0$.

To define the observability notion of interest, consider the sequence of mappings

$$\phi_0(x) = h(x),$$

$$\phi_1(x, v_0) = \frac{\partial \phi_0}{\partial x} [f(x) + g(x)v_0],$$

$$\phi_2(x, v_0, v_1) = \frac{\partial \phi_1}{\partial x}[f(x) + g(x)v_0] + \frac{\partial \phi_1}{\partial v_0} v_1,$$

$$\cdots$$

$$\begin{aligned}
\phi_k(x, v_0, v_1, \cdots, v_{k-1}) &= \frac{\partial \phi_{k-1}}{\partial x} \\
&\times [f(x) + g(x)v_0] \\
&+ \sum_{i=0}^{k-2} \frac{\partial \phi_{k-1}}{\partial v_i} v_{i+1},
\end{aligned} \tag{11}$$

with $k \le n$. Note that if $u(t)$ is of class C^{k-1}, then

$$y^{(k)}(t) = \phi_k(x(t), u(t), \cdots, u^{(k-1)}(t)),$$

where the notation $y^{(k)}(t)$, with k positive integer, is used to denote the k-th order derivative of the function $y(t)$, provided it exists. The mappings ϕ_0 to ϕ_{n-1} can be collected into a unique mapping $\Phi : I\!R^n \times I\!R^{n-1} \to I\!R^n$ defined as

$$\Phi(x, v_0, v_1, \cdots, v_{n-2})$$
$$= \begin{bmatrix} \phi_0(x) \\ \phi_1(x, v_0) \\ \vdots \\ \phi_{n-1}(x, v_0, v_1, \cdots, v_{n-2}) \end{bmatrix}.$$

The mapping Φ is, by construction, such that

$$\Phi(x(t), u(t), \dot{u}(t), \cdots, u^{(n-2)}(t))$$
$$= \begin{bmatrix} y(t) \ \dot{y}(t) \cdots y^{(n-1)}(t) \end{bmatrix}',$$

for any t in which the indicated signals exist. As a consequence, if the mapping Φ is such that, as some point $(\bar{x}, \bar{v})$, where $\bar{v} = \begin{bmatrix} \bar{v}_0 \ \bar{v}_1 \cdots \bar{v}_{n-2} \end{bmatrix}'$,

$$\text{rank}\frac{\partial \Phi}{\partial x}(\bar{x}, \bar{v}) = n, \qquad (12)$$

then, by the Implicit Function Theorem, there exists locally around $(\bar{x}, \bar{v})$ a smooth mapping $\Psi : IR^n \times IR^{n-1} \to IR^n$ such that

$$\omega = \Phi(\Psi(\omega, v), v),$$

i.e., the mapping Ψ is the *inverse* of Φ, parameterized by v. We conclude the discussion noting that if, at a certain time $\bar{t}$, $\bar{x} = x(\bar{t})$ and $\bar{v} = \left[u(\bar{t}) \; \dot{u}(\bar{t}) \cdots u^{(n-2)}(\bar{t}) \right]'$ are such that the rank condition (12) holds, then the mapping Ψ can be used to reconstruct the state of the system from measurements of the input, and its derivatives, and the output and its derivatives, for all t in a neighborhood of $\bar{t}$: $x(t) = \Psi(\omega(t), v(t))$, for all t in a neighborhood of $\bar{t}$, where

$$v(t) = \left[u(t) \; \dot{u}(t) \cdots u^{(n-2)}(t) \right]',$$
$$\omega(t) = \left[y(t) \; \dot{y}(t) \cdots y^{(n-1)}(t) \right]'. \qquad (13)$$

This property is a local property: to derive a property which allows a global reconstruction of the state, we need to impose additional conditions.

Definition 2 Consider the system (10) with $m = p = 1$. The system is said to be *uniformly observable* if:

(i) The mapping $H : IR^n \to IR^n$ defined as

$$H(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{n-1} h(x) \end{bmatrix}$$

is a global diffeomorphism. The functions $L_f^i h$, with i nonnegative integer, are defined as $L_f h(x) = L_f^1 h(x) = \frac{\partial h}{\partial x} f(x)$ and, recursively, as $L_f^{i+1} h(x) = L_f(L_f^i h(x))$.

(ii) The rank condition (12) holds for all $(x, v) \in IR^n \times IR^{n-1}$.

The notion of uniform observability allows to *perform* a global reconstruction of the state, i.e., it makes sure that the identities

$$\omega = \Phi(\Psi(\omega, v), v) \qquad x = \Psi(\Phi(x, v), v)$$

hold for all x, v and ω. In principle, this property may be used in an output feedback control architecture obtained implementing a stabilizing state feedback $u = \alpha(x)$ as $u = \alpha(\Psi(\omega, v))$, with v and ω as given in (13). This implementation is however not possible, since it gives an implicit definition of u and requires the exact differentiation of the input and output signals.

To circumvent these difficulties, one needs to follow a somewhat longer path, as described hereafter. In addition, the global asymptotic stability requirement should be replaced by a less ambitious, yet practically meaningful, requirement: semi-global asymptotic stability. This requirement can be formalized as follows.

Definition 3 The equilibrium x_0 of the system (1), or (7), is said to be semi-globally asymptotically stabilizable if, for each compact set $\mathcal{K} \subset IR^n$ such that $x_0 \in \text{int}(\mathcal{K})$, i.e. the set of all interior points of $\mathcal{K}$, there exists a feedback control law, possibly depending on $\mathcal{K}$, such that the equilibrium x_0 is a locally asymptotically stable equilibrium of the closed-loop system and for any $x(0) \in \mathcal{K}$ one has $\lim_{t \to \infty} x(t) = x_0$.

To bypass the need for the derivatives of the input signal, consider the extended system

$$\dot{x} = f(x) + g(x)v_0, \qquad \dot{v}_0 = v_1,$$
$$\dot{v}_1 = v_2, \quad \cdots \quad \dot{v}_{n-1} = \tilde{u}. \qquad (14)$$

Note that, as described in section "Feedback Systems", if the equilibrium $x_0 = 0$ of the system $\dot{x} = f(x) + g(x)u$ is globally asymptotically stabilizable by a (smooth) feedback $u = \alpha(x)$, then there exists a smooth state feedback $\tilde{u} = \tilde{\alpha}(x, v_0, v_1, \cdots, v_{n-1})$ which globally asymptotically stabilizes the zero equilibrium of the system (14). In the feedback $\tilde{\alpha}$, one can replace x with $\psi(\omega, v)$, thus yielding a feedback of the measurable part of the state of the system (14) and of the output y and its derivatives. Note that if $\omega(t) = \left[y(t) \; \dot{y}(t) \cdots y^{(n-1)}(t) \right]'$, then the feedback $\tilde{u} = \tilde{\alpha}(\psi(\omega, v), v_0, v_1, \cdots, v_{n-1})$ globally asymptotically stabilizes the zero equilibrium of the system (14).

To avoid the computation of the derivatives of y, we exploit the uniform observability property, which implies that the auxiliary system

$$\begin{aligned} \dot{\eta}_0 &= \eta_1, \\ \dot{\eta}_1 &= \eta_2, \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \cdots, v_{n-1}), \end{aligned} \quad (15)$$

with $\eta = \begin{bmatrix} \eta_0, \eta_1, \cdots, \eta_{n-1} \end{bmatrix}'$, has the property of reproducing $y(t)$ and its derivatives up to $y^{(n-1)}(t)$ if properly initialized. This initialization is not feasible, since it requires the knowledge of the derivative of the output at $t = 0$. Nevertheless, the auxiliary system (15) can be modified to provide an *estimate* of y and its derivatives. The modification is obtained adding a linear correction term yielding the system

$$\begin{aligned} \dot{\eta}_0 &= \eta_1 && + L\, c_{n-1}(y - \eta_0), \\ \dot{\eta}_1 &= \eta_2 && + L^2\, c_{n-2}(y - \eta_0), \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \cdots, v_{n-1}) + L^n\, c_0(y - \eta_0), \end{aligned} \quad (16)$$

with $L > 0$ and the coefficients $c_0, \cdots, c_{n-1}$ such that all roots of the polynomial $\lambda^n + c_{n-1}\lambda^{n-1} + c_1\lambda + c_0$ are in C^-. The system (16) has the ability to provide asymptotic estimates of y and its derivatives up to $y^{(n-1)}$ provided these are bounded and the gain L is selected sufficiently large, i.e., the system (16) is a *semi-global* observer of y and its derivatives up to $y^{(n-1)}$.

The closed-loop system obtained using the feedback law $\tilde{u} = \tilde{\alpha}(\psi(\eta, v), v_0, v_1, \cdots, v_{n-1})$ has a locally asymptotically stable equilibrium at the origin. To achieve semi-global stability, one has to select L sufficient large and replace ψ with

$$\tilde{\psi}(\eta, v) = \begin{cases} \psi(\eta, v), & \text{if } \|\psi(\eta, v)\| < M, \\ M\, \dfrac{\psi(\eta, v)}{\|\psi(\eta, v)\|}, & \text{if } \|\psi(\eta, v)\| \geq M, \end{cases}$$

with $M > 0$ to be selected, as detailed in the following statement.

Theorem 3 *Consider the system (10) with $m = p = 1$. Let $x_0 = 0$ be an achievable equilibrium. Assume $h(0) = 0$. Suppose the system is uniformly observable and there exists a smooth state feedback control law $u = \alpha(x)$ which globally asymptotically stabilizes the zero equilibrium and it is such that $\alpha(0) = 0$.*

Then for each $R > 0$, there exist $\tilde{R} > 0$ and $M^\star > 0$, and for each $M > M^\star$, there exists $L^\star$

such that for each $M > M^\star$ and $L > L^\star$, the dynamic output feedback control law

$$\begin{aligned} \dot{v}_0 &= v_1, \\ \dot{v}_1 &= v_2, \\ &\vdots \\ \dot{v}_{n-1} &= \tilde{\alpha}(\tilde{\psi}(\eta, v), v_0, v_1, \cdots, v_{n-1}) \\ \dot{\eta}_0 &= \eta_1 + L\, c_{n-1}(y - \eta_0), \\ \dot{\eta}_1 &= \eta_2 + L^2\, c_{n-2}(y - \eta_0), \\ &\vdots \\ \dot{\eta}_{n-1} &= \phi_n(\psi(\eta, v), v_0, v_1, \cdots, v_{n-1}) \\ &\quad + L^n\, c_0(y - \eta_0), \\ u &= v_0, \end{aligned}$$

yields a closed-loop system with the following properties:
- *The zero equilibrium of the system is locally asymptotically stable.*
- *For any $x(0)$, $v(0)$ and $\eta(0)$ such that $\|x(0)\| < R$ and $\|(v(0), \eta(0))\| < \tilde{R}$, the trajectories of the closed-loop system are such that*

$$\lim_{t \to \infty} x(t) = 0, \qquad \lim_{t \to \infty} v(t) = 0,$$

$$\lim_{t \to \infty} \eta(t) = 0.$$

The foregoing result can be informally formulated as follows: global state feedback stabilizability and uniform observability imply semi-global stabilizability by output feedback. This can be regarded as a nonlinear enhancement of the so-called separation principle for the stabilization, by output feedback, of linear systems. Note, finally, that a global version of the separation principle can be derived under one additional assumption: the existence of an estimator of the norm of the state x.

Summary and Future Directions

A necessary condition and a sufficient condition for stabilizability of an equilibrium of a nonlinear system have been given, together with two systematic design methods. The necessary condition allows to rule out, using a simple algebraic test, existence of continuous stabilizers, whereas the sufficient condition provides a link with classical Lyapunov theory. In addition, the problem of semi-global stability by dynamic output feedback has been discussed in detail. Several issues have not been discussed, including the use of discontinuous, hybrid and time-varying feedbacks; stabilization by static output feedback and dynamic state feedback; robust stabilization. Note finally that similar considerations can be carried out for nonlinear discrete-time systems.

Cross-References

- ▶ Controllability and Observability
- ▶ Fundamental Limitation of Feedback Control
- ▶ Input-to-State Stability
- ▶ Linear State Feedback
- ▶ Lyapunov's Stability Theory
- ▶ Lyapunov Methods in Power System Stability
- ▶ Observers for Nonlinear Systems
- ▶ Observers in Linear Systems Theory
- ▶ Power System Voltage Stability
- ▶ Small Signal Stability in Electric Power Systems
- ▶ Stability and Performance of Complex Systems Affected by Parametric Uncertainty
- ▶ Stability: Lyapunov, Linear Systems

Recommended Reading

Classical references on stabilization for nonlinear systems and on recent research directions are given below.

Bibliography

Artstein Z (1983) Stabilization with relaxed controls. Nonlinear Analysis Theory Methods Appl 7:1163–1173

Astolfi A, Praly L (2003) Global complete observability and output-to-state stability imply the existence of a globally convergent observer. In: Proceedings of the 42nd IEEE conference on decision and control, Maui, pp 1562–1567

Astolfi A, Praly L (2006) Global complete observability and output-to-state stability imply the existence of a globally convergent observer. Math Control Signals Syst 18:32–65

Astolfi A, Karagiannis D, Ortega R (2008) Nonlinear and adaptive control with applications. Springer, London

Bacciotti A (1992) Local stabilizability of nonlinear control systems. World Scientific, Singapore/River Edge

Brockett RW (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometry control theory. Birkhauser, Boston, pp 181–191

Gauthier J-P, Kupka I (2001) Deterministic observation theory and applications. Cambridge University Press, Cambridge/New York

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin

Isidori A (1999) Nonlinear control systems II. Springer, London

Jurdjevic V, Quinn JP (1978) Controllability and stability. J Differ Equ 28:381–389

Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice-Hall, Upper Saddle River

Krstić M, Kanellakopoulos I, Kokotović P (1995) Nonlinear and adaptive control design. Wiley, New York

Marino R, Tomei P (1995) Nonlinear control design: geometric, adaptive and robust. Prentice-Hall, London

Mazenc F, Praly L (1996) Adding integrations, saturated controls, and stabilization for feedforward systems. IEEE Trans Autom Control 41:1559–1578

Mazenc F, Praly L, Dayawansa WP (1994) Global stabilization by output feedback: examples and counterexamples. Syst Control Lett 23:119–125

Ortega R, Loría A, Nicklasson PJ, Sira-Ramírez H (1998) Passivity-based control of Euler-Lagrange systems. Springer, London

Ryan EP (1994) On Brockett's condition for smooth stabilizability and its necessity in a context of nonsmooth feedback. SIAM J Control Optim 32:1597–1604

Sepulchre R, Janković M, Kokotović P (1996) Constructive nonlinear control. Springer, Berlin

Sontag ED (1989) A "universal" construction of Artstein's theorem on nonlinear stabilization. Syst Control Lett 13:117–123

Teel A, Praly L (1995) Tools for semiglobal stabilization by partial state and output feedback. SIAM J Control Optim 33(5):1443–1488

van der Schaft A (2000) L_2-gain and passivity techniques in nonlinear control, 2nd edn. Springer, London

Financial Markets Modeling

Monique Jeanblanc
Laboratoire Analyse et Probabilités, IBGBI,
Université d'Evry Val d'Essonne, Evry Cedex,
France

Abstract

Mathematical finance is an important part of applied mathematics since the 1980s. At the beginning, the main goal was to price derivative products and to provide hedging strategies. Nowadays, the goal is to provide models for prices and interest rates, such that better calibration of parameters can be done. In these pages, we present some basic models. Details can be found in Musiela and Rutkowski (2005).

Keywords

Models for Prices of Stocks

The first model of prices was elaborated by Louis Bachelier, in his thesis (1900). The idea was that the dynamic of prices has a trend, perturbed by a noise. For this noise, Bachelier set the fundamental properties of the Brownian motion. Bachelier's prices were of the form

$$S_t^B = S_0^B + \nu t + \sigma W_t$$

where W is a Brownian motion.

This model, where prices can take negative values, was changed by taking the exponential as in the celebrated Black-Scholes-Merton model (BSM) where

$$S_t = e^{S_t^B} = S_0 \exp(\nu t + \sigma W_t).$$

Only two constant parameters were needed; the coefficient σ is called the volatility. From Itô's formula, the dynamics of the BSM's price are

$$dS_t = S_t(\mu dt + \sigma dW_t)$$

where $\mu = \nu + \frac{1}{2}\sigma^2$. The interest rate is assumed to be a constant r.

The price of a derivative product of payoff $H \in \mathcal{F}_T = \sigma(S_s, s \leq T)$ is obtained using a hedging procedure. One proves that there exists a (self-financing) portfolio with value V, investing in the savings account and in the stock S, with terminal value H, i.e., $V_T = H$. The price of H at time t is V_t. Using that methodology, the price of a European call with strike K (i.e., for $H = (S_T - K)^+$) is

$$V_t = \mathcal{BS}(\sigma, K)_t := S_t \mathcal{N}(d_1) - Ke^{-r(T-t)}\mathcal{N}(d_2)$$

where $\mathcal{N}$ is the cumulative distribution function of a standard Gaussian law and

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left(\ln\left(\frac{S_t}{Ke^{-r(T-t)}}\right)\right)$$
$$+ \frac{1}{2}\sigma\sqrt{T-t}, \ d_2 = d_1 - \sigma\sqrt{T-t}$$

Note that the coefficient μ plays no role in this pricing methodology. This formula opened the door to the notion of risk neutral probability measure: the price of the option (or of any derivative product) is the expectation under the unique probability measure $\mathbb{Q}$, equivalent to $\mathbb{P}$ such that the discounted price $S_t e^{-rt}, t \geq 0$ is a $\mathbb{Q}$ martingale.

During one decade, the financial market was quite smooth and this simple model was efficient to price derivative products and to calibrate the coefficients from the data. After the Black Monday of October 1987, the model was recognized to suffer some weakness and the door was fully

open to more sophisticated models, with more parameters. In particular, the smile effect was seen on the data: the BSM formula being true, the price of the option would be a deterministic function of the fixed parameter σ and of K (the other parameter as maturity, underlying price, interest rate being fixed), and one would obtain, using $\psi(\cdot, K)$, the inverse of $\mathcal{BS}(\cdot, K)$, the constant $\sigma = \psi(C^o(K), K)$, where $C^o(K)$ is the observed prices associated with the strike K. This is not the case, the curve $K \rightarrow \psi(C^o(K), K)$ having a smile shape (or a smirk shape). The BSM model is still used as a benchmark in the concept of implied volatility: for a given observed option price $C^o(K, T)$ (with strike K and maturity T), one can find the value of σ^* such that $\mathcal{BS}(\sigma^*, K, T) = C^o(K, T)$. The surface $\sigma^*(K, T)$ is called the implied volatility surface and plays an important role in calibration issues.

Due to the need of more accurate formula, many models were presented, the only (mathematical) restriction being that prices have to be semi-martingales (to avoid arbitrage opportunities).

A first class is the stochastic volatility models. Assuming that the diffusion coefficient (called the local volatility) is a deterministic function of time and underlying, i.e.,

$$dS_t = S_t(\mu dt + \sigma(t, S_t) dW_t)$$

Dupire proved, using Kolmogorov backward equation that the function σ is determined by the observed prices of call options by

$$\frac{1}{2}K^2\sigma^2(T, K) = \frac{\partial_T C^o(K, T) + rK\partial_K C^o(K, T)}{\partial^2_{KK} C^o(K, T)}$$

where ∂_T (resp. ∂_K) is the partial derivative operator with respect to the maturity (resp., the strike). However, this important model (which allows hedging for derivative products) does not allow a full calibration of the volatility surface.

A second class consists of assuming that the volatility is a stochastic process. The first example is the Heston model which assumes that

$$dS_t = S_t(\mu dt + \sqrt{v_t} dW_t)$$

$$dv_t = -\lambda(v_t - \bar{v})dt + \eta\sqrt{v_t} dB_t$$

where B and W are two Brownian motions with correlation factor ρ. This model is generalized by Gourieroux and Sufana (2003) as Wishart model where the risky asset S is a d dimensional process, with matrix of quadratic variation Σ satisfy

$$dS_t = \text{Diag}(S_t)(\mu dt + \sqrt{\Sigma_t} dW_t)$$
$$d\Sigma_t = (\Lambda\Lambda^T + M\Sigma_t + \Sigma_t M^T)dt$$
$$+ \sqrt{\Sigma_t}\, dB_t\, Q + Q^T\, (dB_t)^T\, \sqrt{\Sigma_t}$$

where W is a d dimensional Brownian motion and B a $(d \times d)$ matrix Brownian, Λ, M, Q are $(d \times d)$ matrices, M is semidefinite negative, and $\Lambda\Lambda^T = \beta QQ^T$ with $\beta \geq d - 1$ to ensure strict positivity. The efficiency of these models was checked using calibration methodology; however, the main idea of hedging for pricing issues is forgotten: there is, in general, no hedging strategy, even for common derivative products, and the validation of the model (the form of the volatility, since the drift term plays no role in pricing) is done by calibration. The risk neutral probability is no more unique.

Interest Rate Models

In the beginning of the 1970s, a specific attention was paid for the interest rate modeling, a constant interest rate being far from the real world.

A first class consists of a dynamic for the instantaneous interest rate r.

Vasisek suggested an Ornstein Uhlenbeck diffusion, i.e., $dr_t = a(b - r_t)dt + \sigma dW_t$, the solution being a Gaussian process of the form

$$r_t = (r_0 - b)e^{-at} + b + \sigma \int_0^t e^{-a(t-u)} dW_u.$$

This model is fully tractable, one of its weakness is that the interest rate can take negative values.

Cox, Ingersoll, and Rubinstein (CIR) studied the square root process

$$dr_t = a(b - r_t)dt + \sigma\sqrt{r_t}dB_t,$$

where $ab > 0$ (so that r is nonnegative). No closed form for r is known; however, closed form for the price of the associated zero coupons is known (see below in Affine Processes).

A second class is the celebrated Heath-Jarrow-Morton model (HJM): the starting point being to model the price of a zero coupon with maturity T (i.e., an asset which pays one monetary unit at time T, these prices being observable) in terms of the instantaneous forward rate $f(t, T)$, as

$$B(t, T) = \exp\left(-\int_t^T f(t, u)du\right).$$

Assuming that

$$df(t, T) = \alpha(t, T)dt + \sigma(t, T)dW_t$$

one finds

$$dB(t, T) = B(t, T)(a(t, T)dt + b(t, T)dW_t)$$

where the relationship between a, b and α, β is known. The instantaneous interest rate is $r_t = f(t, t)$. This model is quite efficient; however, no conditions are known in order that the interest rate is positive.

Models with Jumps

Lévy Processes

Following the idea to produce tractable models to fit the data, many models are now based on Lévy's processes. These models are used for modeling prices as $S_t = e^{X_t}$ where X is a Lévy process. They present a nice feature: even if closed forms for pricing are not known, numerical methods are efficient. One of the most popular is the Carr-Geman-Madan-Yor (CGMY) model which is a Lévy process without Gaussian component and with Lévy density

$$\frac{C}{x^{Y+1}}e^{-Mx}\mathbb{1}_{\{x>0\}} + \frac{C}{|x|^{Y+1}}e^{Gx}\mathbb{1}_{\{x<0\}}$$

with $C > 0$, $M \geq 0$, $G \geq 0$, and $Y < 2$.

These models are often presented as a special case of a change of time: roughly speaking, any semi-martingale is a time-changed Brownian motion, and many examples are constructed as $S_t = B_{A_t}$, where B is a Brownian motion and A an increasing process (the change of time), chosen independent of B (for simplicity) and A a Lévy process (for computational issues).

Affine Processes

Affine models were introduced by Duffie et al. (2003) and are now a standard tool for modeling stock prices, or interest rates. An affine process enjoys the property that, for any affine function g,

$$\mathbb{E}(\exp(uX_T + \int_t^T g(X_s)ds)|\mathcal{F}_t)$$
$$= \exp(\alpha(t, T)X_t + \beta(t, T))$$

where α and β are deterministic solutions of PDEs.

A class of affine processes X ($\mathbb{R}^n$-valued) is the one where

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t + dZ_t$$

where the drift vector b is an affine function of x, the covariance matrix $\sigma(x)\sigma^T(x)$ is an affine function of x, W is an n-dimensional Brownian motion, and Z is a pure jump process whose jumps have a given law ν on $\mathbb{R}^n$ and arrive with intensity $f(X_t)$ where f is an affine function. An example is the CIR model (without jumps), where one can find the price of a zero coupon as

$$\mathbb{E}(\exp -\int_t^T r_s ds|\mathcal{F}_t)$$
$$= \Phi(T - t)\exp[-r_t\Psi(T - t)]$$

where

$$\Psi(s) = \frac{2(e^{\gamma s} - 1)}{(\gamma + a)(e^{\gamma s} - 1) + 2\gamma},$$

$$\Phi(s) = \left(\frac{2\gamma e^{(\gamma+a)\frac{s}{2}}}{(\gamma + a)(e^{\gamma s} - 1) + 2\gamma}\right)^{\frac{2ab}{\sigma^2}},$$

$$\gamma^2 = a^2 + 2\sigma^2.$$

Models for Defaults

At the end of 1990s, new kinds of financial products appear on the market: defaultable options, defaultable zero coupons, and credit derivatives, as the CDOs. Then, particular attention was paid to the modeling of default times; see Bielecki and Rutkowski ([2001](#)). The most popular model for a single default is the reduced form approach, which is based on the knowledge of the intensity rate process. Given a nonnegative process λ, adapted with respect to a reference filtration $\mathbb{F}$, a random time τ is defined so that

$$\mathbb{P}(\tau > t | \mathcal{F}_\infty) = \exp(-\int_0^t \lambda_s ds)$$

This implies that $\mathbb{1}_{\tau \leq t} - \int_0^{t \wedge \tau} \lambda_u du$ is a martingale (in the smallest filtration $\mathbb{G}$ which contains $\mathbb{F}$ and makes τ a random time). This intensity rate λ plays the role of the interest spread due to the equality, for any $Y \in \mathcal{F}_T$ and $\mathbb{F}$ adapted interest rate r

$$\mathbb{E}(Y \mathbb{1}_{T < \tau} \exp(-\int_t^T r_s ds) | \mathcal{G}_t) \mathbb{1}_{t < \tau}$$
$$= \mathbb{1}_{t < \tau} \mathbb{E}(Y \exp(-\int_t^T (r_s + \lambda_s) ds) | \mathcal{F}_t)$$

The real challenge is to model multi-defaults. Defaults are assumed to occur at times τ_i, and one has to describe the (conditional) joint law of the vector $\tau = (\tau_1, \tau_2, \ldots, \tau_n)$, the number n of defaults being quite large (100). A first step is to study the law of the defaults, i.e.,

$$\mathbb{P}(\tau_1 > t_1, \ldots, \tau_n > t_n)$$

which is performed using the classical copula approach, based on the knowledge of the marginal laws of the τ_i, i.e., on the knowledge of $\mathbb{P}(\tau_i \leq s) =: F_i(s)$ where the cumulative distribution functions F_i are assumed invertible. The simplest and most popular copula being the Gaussian one

$$\mathbb{P}(F_i^{-1}(\tau_i) \leq u_i, \ i = 1, \ldots, n)$$
$$= \mathcal{N}_\Sigma^n \left(\mathcal{N}^{-1}(u_1), \ldots, \mathcal{N}^{-1}(u_n) \right),$$

where $\mathcal{N}_\Sigma^n$ is the c.d.f. for the n-variate central normal distribution with the linear correlation matrix Σ and $\mathcal{N}^{-1}$ is the inverse of the c.d.f. for the univariate standard normal distribution. However, a dynamical model was needed, mainly to study the contagion effect, i.e., how the occurrence of a default affect the probability of occurrence of the next defaults.

A first class of models is based on the intensity: starting from a given family of intensities $(\lambda_t^i, i = 1, \ldots, n)$ which satisfies

$$d\lambda_t^i = f(t, \lambda_t^i) dt + g(t, \lambda_t^i) dW_t + dL_t^{(-i)}$$

where $L_t^{(-i)} = \sum_{k \neq i} \beta_k \mathbb{1}_{\tau_k \leq t}$, one constructs random times having the given intensities.

Another class of models, which allows for common jumps, is based on Markov Chains with absorbing state, the default time of the i-th entity being the time where the i-th component of the Markov Chain enters in the absorbing state. These models are efficient due to the introduction of common factors.

Many studies are done to find a form of a dynamic copula, i.e., a family of processes $G_t(\theta), t \geq 0$ such that

$$G_t(\theta) = \mathbb{P}(\tau_1 > \theta_1, \ldots, \tau_n > \theta_n | \mathcal{F}_t)$$

Some examples where, for any fixed t, the quantity $G_t(\cdot)$ is a (stochastic) Gaussian law are known; however, there are few concrete examples for other cases.

Cross-References

▶ Credit Risk Modeling
▶ Option Games: The Interface Between Optimal Stopping and Game Theory

Bibliography

Bachelier L (1900) Théorie de la Spéculation, Thèse, Annales Scientifiques de l'Ecole Normale Supérieure, 21–86, III-17

Bielecki TR, Rutkowski M (2001) Credit risk: modelling valuation and hedging. Springer, Berlin

Duffie D, Filipović D, Schachermayer W (2003) Affine processes and applications in finance. Ann Appl Probab 13:984–1053 (2003)

Gourieroux C, Sufana R (2003) Wishart quadratic term structure, CREF 03–10, HEC Montreal

Musiela M, Rutkowski M (2005) Martingale methods in financial modelling. Springer, Berlin

Flexible Robots

Alessandro De Luca
Sapienza Università di Roma, Roma, Italy

Abstract

Mechanical flexibility in robot manipulators is due to compliance at the joints and/or distributed deflection of the links. Dynamic models of the two classes of robots with flexible joints or flexible links are presented, together with control laws addressing the motion tasks of regulation to constant equilibrium states and of asymptotic tracking of output trajectories. Control design for robots with flexible joints takes advantage of the passivity and feedback linearization properties. In robots with flexible links, basic differences arise when controlling the motion at the joint level or at the tip level.

Keywords

Feedback linearization; Gravity compensation; Joint elasticity; Link flexibility; Noncausal and stable inversion; Regulation by motor feedback; Singular perturbation; Vibration damping

Introduction

Robot manipulators are usually considered as rigid multi-body mechanical systems. This ideal assumption simplifies dynamic analysis and control design but may lead to performance degradation and even unstable behavior, due to the excitation of vibrational phenomena.

Flexibility is mainly due to the limited stiffness of transmissions at the joints (Sweet and Good 1985) and to the deflection of slender and lightweight links (Cannon and Schmitz 1984). Joint flexibility is common when motion transmission/reduction elements such as belts, long shafts, cables, harmonic drives, or cycloidal gears are used. Link flexibility is present in large articulated structures, such as very long arms needed for accessing hostile environments (deep sea or space) or automated crane devices for building construction. In both situations, static displacements and dynamic oscillations are introduced between the driving actuators and the actual position of the robot end effector. Undesired vibrations are typically confined beyond the closed-loop control bandwidth, but flexibility cannot be neglected when large speed/acceleration and high accuracy are requested by the task.

In the dynamic modeling, flexibility is assumed concentrated at the robot joints or distributed along the robot links (most of the times with some finite-dimensional approximation). In both cases, additional generalized coordinates are introduced beside those used to describe the rigid motion of the arm in a Lagrangian formulation. As a result, the number of available control inputs is strictly less than the number of degrees of freedom of the mechanical system. This type of under-actuation, though counterbalanced by the presence of additional potential energy helping to achieve system controllability, suggests that the design of satisfactory motion control laws is harder than in the rigid case.

From a control point of view, different design approaches are needed because of structural differences arising between flexible-joint and flexible-link robots. These differences hold for single- or multiple-link robots, in the linear or nonlinear domain, and depend on the physical co-location or not of mechanical flexibility versus control actuation, as well as on the choice of controlled outputs.

In order to measure the state of flexible robots for trajectory tracking control or feedback stabilization purposes, a large variety of sensing devices can be used, including encoders, joint torque sensors, strain gauges, accelerometers, and high-speed cameras. In particular, measuring the full state of the system would require twice the number of sensors than in the rigid case for robots with flexible joints and possibly more for robots with flexible links. The design of controllers that work provably good with a reduced set of measurements is thus particularly attractive.

Robots with Flexible Joints

Dynamic Modeling

A robot with flexible joints is modeled as an open kinematic chain of $n + 1$ rigid bodies, interconnected by n joints undergoing deflection and actuated by n electrical motors. Let $\boldsymbol{\theta}$ be the n-vector of motor (i.e., rotor) positions, as reflected through the reduction gears, and $\boldsymbol{q}$ the n-vector of link positions. The joint deflection is $\boldsymbol{\delta} = \boldsymbol{\theta} - \boldsymbol{q} \not\equiv \boldsymbol{0}$. The standard assumptions are:

A1 Joint deflections $\boldsymbol{\delta}$ are small, limited to the domain of linear elasticity. The elastic torques due to joint deformations are $\boldsymbol{\tau}_J = \boldsymbol{K}(\boldsymbol{\theta} - \boldsymbol{q})$, where $\boldsymbol{K}$ is the positive definite, diagonal joint stiffness matrix.

A2 The rotors of the electrical motors are modeled as uniform bodies having their center of mass on the rotation axis.

A3 The angular velocity of the rotors is due only to their own spinning.

The last assumption, introduced by Spong (1987), is very reasonable for large reduction ratios and also crucial for simplifying the dynamic model.

From the gravity and elastic potential energy, $\mathcal{U} = \mathcal{U}_g + \mathcal{U}_\delta$, and the kinetic energy $\mathcal{T}$ of the robot, applying the Euler-Lagrange equations to the Lagrangian $\mathcal{L} = \mathcal{T} - \mathcal{U}$ and neglecting all dissipative effects leads to the dynamic model

$$M(q)\ddot{q} + n(q, \dot{q}) + K(q - \theta) = 0 \quad (1)$$

$$B\ddot{\theta} + K(\theta - q) = \tau, \quad (2)$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the positive definite, symmetric inertia matrix of the robot links (including the motor masses); $\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the sum of Coriolis and centrifugal terms $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ (quadratic in $\dot{\boldsymbol{q}}$) and gravitational terms $\boldsymbol{g}(\boldsymbol{q}) = (\partial \mathcal{U}_g / \partial \boldsymbol{q})^T$; $\boldsymbol{B}$ is the positive definite, diagonal matrix of motor inertias (reflected through the gear ratios); and $\boldsymbol{\tau}$ are the motor torques (performing work on $\boldsymbol{\theta}$). The inertia matrix of the *complete* system is then $\mathcal{M}(\boldsymbol{q}) = \text{block diag}\{\boldsymbol{M}(\boldsymbol{q}), \boldsymbol{B}\}$. The two n-dimensional second-order differential equations (1) and (2) are referred to as the *link* and the *motor* equations, respectively. When the joint stiffness $\boldsymbol{K} \to \infty$, it is $\boldsymbol{\theta} \to \boldsymbol{q}$ and $\boldsymbol{\tau}_J \to \boldsymbol{\tau}$, so that the two equations collapse in the limit into the standard dynamic model of rigid robots with total inertia $\mathcal{M}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q}) + \boldsymbol{B}$. On the other hand, when the joint stiffness $\boldsymbol{K}$ is relatively large but still finite, robots with elastic joints show a *two-time-scale* dynamic behavior. A common large scalar factor $1/\epsilon^2 \gg 1$ can be extracted from the diagonal stiffness matrix as $\boldsymbol{K} = \hat{\boldsymbol{K}}/\epsilon^2$. The *slow* subsystem is associated to the link dynamics

$$M(q)\ddot{q} + n(q, \dot{q}) = \tau_J, \quad (3)$$

while the *fast* subsystem takes the form

$$\epsilon^2 \ddot{\boldsymbol{\tau}}_J = \hat{\boldsymbol{K}} \left(\boldsymbol{B}^{-1}(\boldsymbol{\tau} - \boldsymbol{\tau}_J) \right. \\ \left. + \boldsymbol{M}^{-1}(\boldsymbol{q})(\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\tau}_J) \right) \quad (4)$$

For small ϵ, Eqs. (3) and (4) represent a singularly perturbed system. The two separate time scales governing the slow and fast dynamics are t and $\sigma = t/\epsilon$.

Regulation

The basic robotic task of moving between two arbitrary equilibrium configurations is realized by a feedback control law that asymptotically stabilizes the desired robot state.

In the absence of gravity ($\boldsymbol{g} \equiv \boldsymbol{0}$), the equilibrium states are parameterized by the desired reference position $\boldsymbol{q}_d$ of the links and take the

form $q = q_d$, $\theta = \theta_d = q_d$ (with no joint deflection at steady state) and $\dot{q} = \dot{\theta} = 0$. As a result of passivity of the mapping from τ to $\dot{\theta}$, global regulation is achieved by a *decentralized PD* law using only feedback from the motor variables,

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta}, \qquad (5)$$

with diagonal $K_P > 0$ and $K_D > 0$.

In the presence of gravity, the (unique) equilibrium position of the motor associated with a desired link position q_d becomes $\theta_d = q_d + K^{-1}g(q_d)$. Global regulation is obtained by adding an extra gravity-dependent term τ_g to the PD control law (5),

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + \tau_g, \qquad (6)$$

with diagonal matrices $K_P > 0$ (at least) and $K_D > 0$. The term τ_g needs to match the gravity load $g(q_d)$ at steady state. The following choices are of slight increasing control complexity, with progressively better transient performance.

- *Constant* gravity compensation: $\tau_g = g(q_d)$. Global regulation is achieved when the smallest positive gain in the diagonal matrix K_P is large enough (Tomei 1991). This sufficient condition can be enforced only if the joint stiffness K dominates the gradient of gravity terms.
- *Online* compensation: $\tau_g = g(\tilde{\theta})$, $\tilde{\theta} = \theta - K^{-1}g(q_d)$. Gravity effects on the links are approximately compensated during robot motion. Global regulation is proven under the same conditions above (De Luca et al. 2005).
- *Quasi-static* compensation: $\tau_g = g(\tilde{q}(\theta))$. At any measured motor position θ, the link position $\tilde{q}(\theta)$ is computed by solving numerically $g(q) + K(q - \theta) = 0$. This removes the need of a strictly positive lower bound on K_P (Kugi et al. 2008), but the joint stiffness should still dominate the gradient of gravity terms.

Additional feedback from the full robot state $(q, \dot{q}, \theta, \dot{\theta})$, measured or reconstructed through dynamic observers, can provide faster and damped transient responses. This solution is particularly convenient when a joint torque sensor measuring τ_J is available (*torque-controlled* robots). Using

$$\tau = K_P(\theta_d - \theta) - K_D\dot{\theta} + K_T(g(q_d) - \tau_J)$$
$$- K_S\dot{\tau}_J + g(q_d), \qquad (7)$$

the four diagonal gain matrices can be given a special structure so that asymptotic stability is automatically guaranteed (Albu-Schäffer and Hirzinger 2001).

Trajectory Tracking

Let a desired sufficiently smooth trajectory $q_d(t)$ be specified for the robot links over a finite or infinite time interval. The control objective is to asymptotically stabilize the trajectory tracking error $e = q_d(t) - q(t)$ to zero, starting from a generic initial robot state. Assuming that $q_d(t)$ is four times continuously differentiable, a torque input profile $\tau_d(t) = \tau_d(q_d, \dot{q}_d, \ddot{q}_d, \dddot{q}_d, \ddddot{q}_d)$ can be derived from the dynamic model (1) and (2) so as to reproduce exactly the desired trajectory, when starting from matched initial conditions. A local solution to the trajectory tracking problem is provided by the combination of such feedforward term $\tau_d(t)$ with a stabilizing linear feedback from the partial or full robot state; see Eqs. (6) or (7).

When the joint stiffness is large enough, one can take advantage of the system being *singularly perturbed*. A control law τ_s designed for the rigid robot will deal with the slow dynamics, while a relatively simple action τ_f is used to stabilize the fast vibratory dynamics around an invariant manifold associated to the rigid robot control (Spong et al. 1987). This class of composite control laws has the general form

$$\tau = \tau_s(q, \dot{q}, t) + \epsilon\tau_f(q, \dot{q}, \tau_J, \dot{\tau}_J). \qquad (8)$$

When setting $\epsilon = 0$ in Eqs. (3), (4), and (8), the control setup of the equivalent rigid robot is recovered as

$$(M(q) + B)\ddot{q} + n(q, \dot{q}) = \tau_s. \qquad (9)$$

Though more complex, the best performing trajectory tracking controller for the general case is based on *feedback linearization*. Spong (1987) has shown that the nonlinear state feedback

$$\boldsymbol{\tau} = \boldsymbol{\alpha}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \dddot{\boldsymbol{q}}) + \boldsymbol{\beta}(\boldsymbol{q})\boldsymbol{v}, \qquad (10)$$

with

$$\begin{aligned}
\boldsymbol{\alpha} =\ & \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\
& + \boldsymbol{B}\boldsymbol{K}^{-1}\left(\left(\ddot{\boldsymbol{M}}(\boldsymbol{q}) + \boldsymbol{K}\right)\ddot{\boldsymbol{q}} + 2\dot{\boldsymbol{M}}(\boldsymbol{q})\dddot{\boldsymbol{q}}\right. \\
& \left. + \ddot{\boldsymbol{n}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right) \\
\boldsymbol{\beta} =\ & \boldsymbol{B}\boldsymbol{K}^{-1}\boldsymbol{M}(\boldsymbol{q}),
\end{aligned}$$

leads globally to the closed-loop linear system

$$\boldsymbol{q}^{[4]} = \boldsymbol{v}, \qquad (11)$$

i.e., to decoupled chains of four input–output integrators from each auxiliary input v_i to each link position output q_i, for $i = 1, \ldots, n$. The control design is then completed on the linear SISO side, by forcing the trajectory tracking error to be *exponentially stable* with an arbitrary decaying rate. The control law (10) is expressed as a function of the *linearizing coordinates* $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \dddot{\boldsymbol{q}})$ (up to the link jerk), which can be however rewritten in terms of the original state $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ using the dynamic model equations. This fundamental result is the direct extension of the so-called "computed torque" method for rigid robots.

Robots with Flexible Links

Dynamic Modeling

For the dynamic modeling of a single flexible link, the distributed nature of structural flexibility can be captured, under suitable assumptions, by partial differential equations (PDE) with associated boundary conditions. A common model is the *Euler-Bernoulli* beam. The link is assumed to be a slender beam, with uniform geometric characteristics and homogeneous mass distribution, clamped at the base to the rigid hub of an actuator producing a torque τ and rotating on a horizontal plane. The beam is flexible in the lateral direction only, being stiff with respect to axial forces, torsion, and bending due to gravity. Deformations are small and are in the elastic domain. The physical parameters of interest are the linear density ρ of the beam, its flexural rigidity EI, the beam length ℓ, and the hub inertia I_h (with $I_t = I_h + \rho\ell^3/3$). The equations of motion combine lumped and distributed parameter parts, with the hub rotation $\theta(t)$ and the link deformation $w(x, t)$, being $x \in [0, \ell]$ the position along the link. From Hamilton principle, we obtain

$$I_t\,\ddot{\theta}(t) + \rho\int_0^\ell x\,\ddot{w}(x, t)\,dx = \tau(t) \qquad (12)$$

$$EI\,w''''(x, t) + \rho\,\ddot{w}(x, t) + \rho\,x\,\ddot{\theta}(t) = 0 \quad (13)$$

$$\begin{aligned}
w(0, t) &= w'(0, t) = 0, \\
w''(\ell, t) &= w'''(\ell, t) = 0, \qquad (14)
\end{aligned}$$

where a prime denotes partial derivative w.r.t. to space. Equation (14) are the clamped-free boundary conditions at the two ends of the beam (no payload is present at the tip).

For the analysis of this self-adjoint PDE problem, one proceeds by separation of variables in space and time, defining

$$w(x, t) = \phi(x)\delta(t) \qquad \theta(t) = \alpha(t) + k\delta(t), \tag{15}$$

where $\phi(x)$ is the link spatial deformation, $\delta(t)$ is its time behavior, $\alpha(t)$ describes the angular motion of the instantaneous center of mass of the beam, and k is chosen so as to satisfy (12) for $\tau = 0$. Being system (12)–(14) linear, nonrational transfer functions can be derived in the Laplace transform domain between the input torque and some relevant system output, e.g., the angular position of the hub or of the tip of the beam (Kanoh 1990). The PDE formalism provides also a convenient basis for analyzing distributed sensing, feedback from strain sensors (Luo 1993), or even distributed actuation with piezo-electric devices placed along the link.

The transcendental characteristic equation associated to the spatial part of the solution to Eqs. (12)–(14) is

$$I_h \gamma^3 \big(1 + \cos(\gamma \ell) \cosh(\gamma \ell)\big)$$
$$+ \rho \big(\sin(\gamma \ell) \cosh(\gamma \ell) - \cos(\gamma \ell) \sinh(\gamma \ell)\big) = 0. \tag{16}$$

When the hub inertia $I_h \to \infty$, the second term can be neglected and the characteristic equation collapses into the so-called *clamped* condition. Equation (16) has an infinite but countable number of positive real roots γ_i, with associated eigenvalues of resonant frequencies $\omega_i = \gamma_i^2 \sqrt{EI/\rho}$ and orthonormal eigenvectors $\phi_i(x)$, which are the natural deformation shapes of the beam (Barbieri and Özgüner 1988). A finite-dimensional dynamic model is obtained by truncation to a finite number m_e of eigenvalues/shapes. From

$$w(x,t) = \sum_{i=1}^{m_e} \phi_i(x) \delta_i(t) \tag{17}$$

we get

$$I_t \ddot{\alpha}(t) = \tau(t)$$
$$\ddot{\delta}_i(t) + \omega_i^2 \delta_i(t) = \phi_i'(0) \tau(t), \tag{18}$$
$$i = 1, \ldots, m_e,$$

where the rigid body motion (top equation) appears as decoupled from the flexible dynamics, thanks to the choice of variable α rather than θ. Modal damping can be added on the left-hand sides of the lower equations through terms $2\zeta_i \omega_i \dot{\delta}_i$ with $\zeta_i \in [0,1]$. The angular position of the motor hub at the joint is given by

$$\theta(t) = \alpha(t) + \sum_{i=1}^{m_e} \phi_i'(0) \delta_i(t), \tag{19}$$

while the tip angular position is

$$y(t) = \alpha(t) + \sum_{i=1}^{m_e} \frac{\phi_i(\ell)}{\ell} \delta_i(t). \tag{20}$$

The *joint-level* transfer function $p_{\text{joint}}(s) = \theta(s)/\tau(s)$ will always have relative degree two and only minimum phase zeros. On the other hand, the *tip-level* transfer function $p_{\text{tip}}(s) = y(s)/\tau(s)$ will contain non-minimum phase zeros. This basic difference in the pattern of the transmission zeros is crucial for motion control design.

In a simpler modeling technique, a specified class of spatial functions $\phi_i(x)$ is assumed for describing link deformation. The functions need to satisfy only a reduced set of geometric boundary conditions (e.g., *clamped* modes at the link base), but otherwise no dynamic equations of motion such as (13). The use of finite-dimensional expansions like (17) limits the validity of the resulting model to a maximum frequency. This truncation must be accompanied by suitable filtering of measurements and of control commands, so as to avoid or limit spillover effects (Balas 1978).

In the dynamic modeling of robots with n flexible links, the resort to *assumed modes* of link deformation becomes unavoidable. In practice, some form of approximation and a finite-dimensional treatment is necessary. Let $\boldsymbol{\theta}$ be the n-vector of joint variables describing the rigid motion, and $\boldsymbol{\delta}$ be the m-vector collecting the deformation variables of all flexible links. Following a Lagrangian formulation, the dynamic model with clamped modes takes the general form (Book 1984)

$$\begin{pmatrix} \boldsymbol{M}_{\theta\theta}(\boldsymbol{\theta}, \boldsymbol{\delta}) & \boldsymbol{M}_{\theta\delta}(\boldsymbol{\theta}, \boldsymbol{\delta}) \\ \boldsymbol{M}_{\theta\delta}^T(\boldsymbol{\theta}, \boldsymbol{\delta}) & \boldsymbol{M}_{\delta\delta}(\boldsymbol{\theta}, \boldsymbol{\delta}) \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{\theta}} \\ \ddot{\boldsymbol{\delta}} \end{pmatrix}$$
$$+ \begin{pmatrix} \boldsymbol{n}_{\theta}(\boldsymbol{\theta}, \boldsymbol{\delta}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\delta}}) \\ \boldsymbol{n}_{\delta}(\boldsymbol{\theta}, \boldsymbol{\delta}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\delta}}) \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{D}\dot{\boldsymbol{\delta}} + \boldsymbol{K}\boldsymbol{\delta} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau} \\ \boldsymbol{0} \end{pmatrix}, \tag{21}$$

where the positive definite, symmetric inertia matrix $\mathcal{M}$ of the complete robot and the Coriolis, centrifugal, and gravitational terms $\boldsymbol{n}$ have been partitioned in blocks of suitable dimensions, $\boldsymbol{K} > 0$ and $\boldsymbol{D} \geq 0$ are the robot link stiffness and damping matrices, and $\boldsymbol{\tau}$ is the n-vector of actuating torques.

F

The dynamic model (21) shows the general couplings existing between nonlinear rigid body motion and linear flexible dynamics. In this respect, the linear model (18) of a single flexible link is a remarkable exception.

The choice of specific assumed modes may simplify the blocks of the robot inertia matrix, e.g., orthonormal modes used for each link induce a decoupled structure of the diagonal inertia subblocks of $M_{\delta\delta}$. Quite often the total kinetic energy of the flexible robot is evaluated only in the undeformed configuration $\delta = 0$. With this approximation, the inertia matrix becomes independent of δ, and so the velocity terms in the model. Furthermore, due to the hypothesis of small deformation of each link, the dependence of the gravity term in the lower component n_δ is only a function of θ.

The validation of (21) goes through the experimental identification of the relevant dynamic parameters. Besides those inherited from the rigid case (mass, inertia, etc.), also the set of structural resonant frequencies and associated deformation profiles should be identified.

Control of Joint-Level Motion

When the target variables to be controlled are defined at the joint level, the control problem for robots with flexible links is similar to that of robots with flexible joints. As a matter of fact, the models (1), (2), and (21) are both *passive* systems with respect to the output θ; see (19) in the scalar case. For instance, regulation is achieved by a PD action with constant gravity compensation, using a control law of the form (6) without the need of feeding back link deformation variables (De Luca and Siciliano 1993a). Similarly, stable tracking of a joint trajectory $\theta_d(t)$ is obtained by a singular perturbation control approach, with flexible modes dynamics acting at multiple time scales with respect to rigid body motion (Siciliano and Book 1988), or by an inversion-based control (De Luca and Siciliano 1993b), where input–output (rather than full state) exact linearization is realized and the effects of link flexibility are canceled on the motion of the robot joints. While vibrational behavior will still affect the robot at the level of end-effector motion, the closed-loop dynamics of the δ variables is stable and link deformations converge to a steady-state constant value (zero in the absence of gravity) thanks to the intrinsic damping of the mechanical structure. Improved transients are indeed obtained by active modal damping control (Cannon and Schmitz 1984).

A control approach specifically developed for the rest-to-rest motion of flexible mechanical systems is *command shaping* (Singer and Seering 1990). The original command designed to achieve a desired motion for a rigid robot is convolved with suitable signals delayed in time, so as to cancel (or reduce to a minimum) the effects of the excited vibration modes at the time of motion completion. For a single slewing link with linear dynamics, as in (18), the rest-to-rest input command is computed in closed form by using impulsive signals and can be made robust via an over-parameterization.

Control of Tip-Level Motion

The design of a control law that allows asymptotic tracking of a desired trajectory for the end effector of a robot with flexible links needs to face the unstable zero dynamics associated to the problem. In the linear case of a single flexible link, this is equivalent to the presence of non-minimum phase zeros in the transfer function to the tip output (20). Direct inversion of the input–output map leads to instability, due to cancellation of non-minimum phase zeros by unstable poles, with link deformation growing unbounded and control saturations.

The solution requires instead to determine the unique reference state trajectory of the flexible structure that is associated to the desired tip trajectory and has *bounded* deformation. Based on regulation theory, the control law will be the superposition of a nominal feedforward action, which keeps the system along the reference state trajectory (and thus the output on the desired trajectory), and of a stabilizing feedback that reduces the error with respect to this state trajectory to zero without resorting to dangerous cancellations.

In general, computing such a control law requires the solution of a set of nonlinear partial

differential equations. However, in the case of a single flexible link with linear dynamics, the feedforward profile is simply derived by an inversion defined in the *frequency domain* (Bayo 1987). The desired tip acceleration $\ddot{y}_d(t)$, $t \in [0, T]$, is considered as part of a rest-to-rest periodic signal, with zero mean value and zero integral. The procedure, implemented efficiently using Fast Fourier Transform on discrete-time samples, will automatically generate bounded time signals only. The resulting unique torque profile $\tau_d(t)$ will be a *noncausal* command, anticipating the actual start of the output trajectory at $t = 0$ (so as to *precharge* the link to the correct initial deformation) and ending after $t = T$ (to *discharge* the residual link deformation and recover the final rest configuration).

The same result was recovered by Kwon and Book (1994) in the time domain, by forward integrating in time the stable part of the inverse system dynamics and backward integrating the unstable part. An extension to the multi-link non-linear case uses an iterative approach on repeated linear approximations of the system along the nominal trajectory (Bayo et al. 1989).

Summary and Future Directions

The presence of mechanical flexibility in the joints and the links of multi-dof robots poses challenging control problems. Control designs take advantage or are limited by some system-level properties. Robots with flexible joints are passive systems at the level of motor outputs, have no zero dynamics associated to the link position outputs, and are always feedback linearizable systems. Robots with flexible links are still passive for joint-level outputs, but cannot be feedback linearized in general, and have unstable zero dynamics (non-minimum phase zeros in the linear case) when considering the end-effector position as controlled output.

State-of-the-art control laws address regulation and trajectory tracking tasks in a satisfactory way, at least in nominal conditions and under full-state feedback. Current research directions are aimed at achieving robustness to model uncertainties and external disturbances (with adaptive, learning, or iterative schemes), and further exploit the design of control laws under limited measurements and noisy sensing. Beyond free motion tasks, an accurate treatment of interaction tasks with the environment, requiring force or impedance controllers, is still missing for flexible robots. In this respect, passivity-based control approaches that do not necessarily operate dynamic cancellations may take advantage of the existing compliance, trading off between improved energy efficiency and some reduction in nominal performance.

Often seen as a limiting factor for performance, the presence of joint elasticity is now becoming an explicit advantage for safe physical human-robot interaction and for locomotion. Next generation lightweight robots and humanoids will use flexible joints and also compact actuation with online controlled variable joint stiffness, an area of active research.

Cross-References

▶ Feedback Linearization of Nonlinear Systems
▶ Modeling of Dynamic Systems from First Principles
▶ Nonlinear Zero Dynamics
▶ PID Control
▶ Regulation and Tracking of Nonlinear Systems

Recommended Reading

In addition to the works cited in the body of this article, a detailed treatment of dynamic modeling and control issues for flexible robots can be found in De Luca and Book (2008). This includes also the use of dynamic feedback linearization for a more general model of robots with elastic joints. For the same class of robots, Brogliato et al. (1995) provided a comparison of passivity-based and inversion-based tracking controllers.

Bibliography

Albu-Schäffer A, Hirzinger G (2001) A globally stable state feedback controller for flexible joint robots. Adv Robot 15(8):799–814

Balas MJ (1978) Feedback control of flexible systems. IEEE Trans Autom Control 23(4):673–679

Barbieri E, Özgüner Ü (1988) Unconstrained and constrained mode expansions for a flexible slewing link. ASME J Dyn Syst Meas Control 110(4):416–421

Bayo E (1987) A finite-element approach to control the end-point motion of a single-link flexible robot. J Robot Syst 4(1):63–75

Bayo E, Papadopoulos P, Stubbe J, Serna MA (1989) Inverse dynamics and kinematics of multi-link elastic robots: an iterative frequency domain approach. Int J Robot Res 8(6):49–62

Book WJ (1984) Recursive Lagrangian dynamics of flexible manipulators. Int J Robot Res 3(3):87–106

Brogliato B, Ortega R, Lozano R (1995) Global tracking controllers for flexible-joint manipulators: a comparative study. Automatica 31(7):941–956

Cannon RH, Schmitz E (1984) Initial experiments on the end-point control of a flexible one-link robot. Int J Robot Res 3(3):62–75

De Luca A, Book W (2008) Robots with flexible elements. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 287–319

De Luca A, Siciliano B (1993a) Regulation of flexible arms under gravity. IEEE Trans Robot Autom 9(4):463–467

De Luca A, Siciliano B (1993b) Inversion-based nonlinear control of robot arms with flexible links. AIAA J Guid Control Dyn 16(6):1169–1176

De Luca A, Siciliano B, Zollo L (2005) PD control with on-line gravity compensation for robots with elastic joints: theory and experiments. Automatica 41(10):1809–1819

Kanoh H (1990) Distributed parameter models of flexible robot arms. Adv Robot 5(1):87–99

Kugi A, Ott C, Albu-Schäffer A, Hirzinger G (2008) On the passivity-based impedance control of flexible joint robots. IEEE Trans Robot 24(2):416–429

Kwon D-S, Book WJ (1994) A time-domain inverse dynamic tracking control of a single-link flexible manipulator. ASME J Dyn Syst Meas Control 116(2):193–200

Luo ZH (1993) Direct strain feedback control of flexible robot arms: new theoretical and experimental results. IEEE Trans Autom Control 38(11):1610–1622

Siciliano B, Book WJ (1988) A singular perturbation approach to control of lightweight flexible manipulators. Int J Robot Res 7(4):79–90

Singer N, Seering WP (1990) Preshaping command inputs to reduce system vibration. ASME J Dyn Syst Meas Control 112(1):76–82

Spong MW (1987) Modeling and control of elastic joint robots. ASME J Dyn Syst Meas Control 109(4):310–319

Spong MW, Khorasani K, Kokotovic PV (1987) An integral manifold approach to the feedback control of flexible joint robots. IEEE J Robot Autom 3(4):291–300

Sweet LM, Good MC (1985) Redefinition of the robot motion control problem. IEEE Control Syst Mag 5(3):18–24

Tome P (1991) A simple PD controller for robots with elastic joints. IEEE Trans Autom Control 36(10):1208–1213

Flocking in Networked Systems

Ali Jadbabaie
University of Pennsylvania, Philadelphia, PA, USA

Abstract

Flocking is a collective behavior exhibited by many animal species such as birds, insects, and fish. Such behavior is generated by distributed motion coordination through nearest-neighbor interactions. Empirical study of such behavior has been an active research in ecology and evolutionary biology. Mathematical study of such behaviors has become an active research area in a diverse set of disciplines, ranging from statistical physics and computer graphics to control theory, robotics, opinion dynamics in social networks, and general theory of multiagent systems. While models vary in detail, they are all based on local diffusive dynamics that results in emergence of consensus in direction of motion. Flocking is closely related to the notion of consensus and synchronization in multiagent systems, as examples of collective phenomena that emerge in multiagent systems as result of local nearest-neighbor interactions.

Keywords

Consensus; Dynamics; Flocking; Graph theory; Markov chains; Switched dynamical systems; Synchronization

Flocking or social aggregation is a group behavior observed in many animal species, ranging

from various types of birds to insects and fish. The phenomena can be loosely defined as any aggregate collective behavior in parallel rectilinear formation or (in case of fish) in collective circular motion. The mechanisms leading to such behavior have been (and continues to be) an active area of research among ecologists and evolutionary biologists, dating back to the 1950s if not earlier. The engineering interest in the topic is much more recent.

In 1986, Craig Reynolds (1987), a computer graphics researcher, developed a computer model of collective behavior for animated artificial objects called boids. The flocking model for boids was used to realistically duplicate aggregation phenomena in fish flocks and bird schools for computer animation. Reynolds developed a simple, intuitive physics-based model: each boid was a point mass subject to three simple steering forces: alignment (to steer each boid towards the average heading of its *local* flockmates), cohesion (steering to move towards the average position of *local* flockmates), and separation (to avoid crowding local flockmates). The term local should be understood as those flockmates who are within each other's influence zone, which could be a disk (or a wide-angle sector of a disk) centered at each boid with a prespecified radius. This simple zone-based model created very realistic flocking behaviors and was used in many animations (Fig. 1). Reynolds' 3 rules of flocking.

Nine years later, in 1995, Vicsek et al. (1995) and coauthors independently developed a model for velocity alignment of self-propelled particles (SPPs) in a square with periodic boundary conditions. SPPs are essentially kinematic particles moving with constant speed and the steering law determines the angle of (what control theorists call a kinematic, nonholonomic vehicle model). Vicsek et al.'s steering law was very intuitive and simple and was essentially Reynolds' zone-based alignment rule (Vicsek was not aware of Reynolds' result): each particle averages the *angle* of its velocity vector with that of its neighbors (those within a disk of a prespecified distance), plus a noise term used to model inaccuracies in averaging. Once the velocity vector is determined at each time, each particle takes a unit step along that direction, then determining its neighbors again and repeating the protocol.

The simulations were done in a square of unit length with periodic boundary conditions, to simulate infinite space. Vicsek and coauthors simulated this behavior and found that as the density of the particles increased, a preferred direction spontaneously emerged, resulting in a global group behavior with nearly aligned velocity vectors for all particles, despite the fact that the update protocol is entirely local.

With the interest in control theory shifting towards multiagent systems and networked coordination and control, it became clear that the mathematics of how birds flock and fish school and how individuals in a social network reach agreement (even though they are often only influenced by other like-minded individuals) are quite related to the question of how can one engineer a swarm of robots to behave like bird flocks.
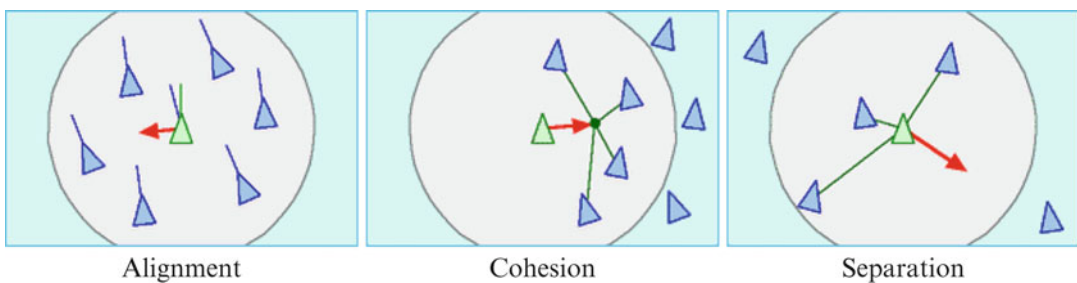
Flocking in Networked Systems, Fig. 1 Photo from http://red3d.com/cwr/boids/

These questions have occupied the minds of many researchers in diverse areas ranging from control theory to robotics, mathematics, and computer science. As discussed above, most of the early research, which happened in computer graphics and statistical physics, was on modeling and simulation of collective behavior. Over the past 13 years, however, the focus has shifted to rigorous systems theoretic foundations, leading to what one might call a theory of collective phenomena in multiagent systems. This theory blends dynamical systems, graph theory, Markov chains, and algorithms.

This type of collective phenomena are often modeled as many-degrees-of-freedom (discrete-time or continuous-time) dynamical systems with an additional twist that the interconnection structure between individual dynamical systems changes, since the motion of each node in a flock (or opinion of an individual) is affected primarily by those in each node's *local neighborhood*. The twist here is that the local neighborhood is not fixed: neighbors are defined based on the actual state of the system, for example, in case of Vicsek's alignment rule, as each particle averages its velocity direction with that of its neighbors and then takes a step, the set of its nearest neighbors can change.

Interestingly, very similar models were developed in statistics and mathematical sociology literature to describe how individuals in a social network update their opinions as a function of the opinion of their friends. The first such model goes back to the seminal work of DeGroot (1974) in 1974. DeGroot's model simply described the evolution of a group's scalar opinion as a function of the opinion of their neighbors by an iterative averaging scheme that can be conveniently modeled as a Markov chain. Individuals are represented by nodes of a graph, start from an opinion at time zero, and then are influenced by the people in their social clique. In DeGroot's model though, the network is given exogenously and does not change as a function of the opinions. The model therefore can be analyzed using the celebrated Perron-Frobenius theorem. The evolution of opinions is a discrete dynamic system that corresponds to an averaging map. When the network is fixed and connected (i.e., there is a path from every node to every other node) and agents also include their own opinions in the averaging, the update results in computation of a global weighted average of initial opinions, where the weight of each initial opinion in the final aggregate is proportional to the "importance" of each node in the network.

The flocking models of Reynolds (1987) and Vicsek et al. (1995), however, have an extra twist: the network changes as opinions are updated. Similarly, more refined sociological models developed over the past decade also capture this endogeneity (Hegselmann and Krause 2002): each individual agent is influenced by others only when their opinion is close to her own. In other words, as opinions evolve, neighborhood structures change as the function of the evolving opinion, resulting in a switched dynamical system in which switching is state dependent.

In a paper in 2003, Jadbabaie and coauthors (2003) studied the Reynolds' alignment rule in the context of Vicsek's model when there is no exogenous noise. To model the endogeneity of the change in neighborhood structure, they developed a model based on repeated local averaging in which the neighborhood structure changes over time and therefore instead of a simple discrete-time linear dynamical system, the model is a discrete linear inclusion or a switched linear system. The question of interest was to determine what regimes of network changes could result in flocking. Clearly, as also DeGroot's model suggests, when the local neighbor structures do not change, connectivity is the key factor for flocking. This is a direct consequence of Perron-Frobenius theory. The result can also be described in terms of directed graphs. What is the equivalent condition in changing networks? Jadbabaie and coauthors show in their paper that indeed connectivity is important, but it need not hold every time: rather, there needs to be time periods over which the graphs are *connected in time*. More formally, the process of neighborhood changes due to motion in Vicsek's model can be simply abstracted as a graph in which the links "blink" on and off. For flocking, one needs to ensure that there are

time periods over which the union of edges (that occur as a result of proximity of particles) needs to correspond to a connected graph and such intervals need to occur infinitely often. It turns out that many of these ideas were developed much earlier in a thesis and following paper by Tsitsiklis (1984) and Tsitsiklis et al. (1986), in the context of distributed and asynchronous computation of global averages in changing graphs, and in a paper by Chatterjee and Seneta (1977), in the context of nonhomogeneous Markov chains. The machinery for proving such a results, however, is classical and has a long history in the theory of inhomogeneous Markov chains, a subject studied since the time of Markov himself, followed by Birkhoff and other mathematicians such as Hajnal, Dobrushin, Seneta, Hatfield, Daubachies, and Lagarias, to name a few.

The interesting twist in analysis of Vicsek's model is that the Euclidean norm of the distance to the globally converged "consensus angle" (or consensus opinion in the case of opinion models) can actually *grow* in a single step of the process; therefore, standard quadratic Lyapunov function arguments which serve as the main tool for analysis of switched linear systems are not suitable for the analysis of such models. However, it is fairly easy to see that under the process of local averaging, the largest value cannot increase and the smallest value cannot decrease. In fact, one can show that if enough connected graphs occur as the result of the switching process, the maximum value will be strictly decreasing and the minimum value will be strictly increasing The paper by Jadbabaie and coauthors (2003) has lead to a flurry of results in this area over the past decade. One important generalization to the results of Jadbabaie et al. (2003), Tsitsiklis (1984), and Tsitsiklis et al. (1986) came 2 years later in a paper by Moreau (2005), who showed that these results can be generalized to nonlinear updates and directed graphs. Moreau showed that any dynamic process that assigns a point in the interior of the convex hull of the value of each node and its neighbors will eventually result in agreement and consensus, if and only if the union of graphs from every time step till infinity

contains a directed spanning tree (a node who has direct links to every other node).

Some of these results were also extended to the analysis of the Reynolds' model of flocking including the other two behaviors. First, Tanner and coauthors (2003, 2007) showed in a series of papers in 2003 and 2007 that a zone-based model similar to Reynolds can result in flocking for dynamic agents, provided that the graph representing interagent communications stays connected. Olfati-Saber and coauthors (2007) developed similar results with a slightly different model.

Many generalizations and extension for these results exist in a diverse set of disciplines, resulting in a rich theory which has had applications from robotics (such as rendezvous in mobile robots) (Cortés et al. 2006) to mathematical sociology (Hegselmann and Krause 2002) and from economics (Golub and Jackson 2010) to distributed optimization theory (Nedic and Ozdaglar 2009). However, some of the fundamental mathematical questions related to flocking still remain open.

First, most results focus on endogenous models of network change. A notable extension is a paper by Cucker and Smale (2007), in which the authors develop and analyze an endogenous model of flocking that cleverly *smoothens out* the discontinuous change in network structure by allowing each node's influence to decay smoothly as a function of distance.

Recently, in a series of papers, Chazelle has made progress in this arena by using tools from computational geometry and algorithms for analysis of endogenous models of flocking (Chazelle 2012). Chazelle has introduced the notion of the *s-energy* of a flock, which can be thought of as a parameterized family of Lyapunov functions that represent the evolution of global misalignment between flockmates. Via tools from dynamical systems, computational geometry, combinatorics, complexity theory, and algorithms, Chazelle creates an "algorithmic calculus," for *diffusive influence systems:* surprisingly, he shows that the orbit or flow of such systems is attracted to a fixed point in the case of undirected graphs and a limit cycle for almost all arbitrarily small random

perturbations. Furthermore, the convergence time can also be bounded in both cases and the bounds are essentially optimal. The setup of the diffusive influence system developed by Chazelle creates a near-universal setup for analyzing various problems involving collective behavior in networked multiagent systems, from flocking, opinion dynamics, and information aggregation to synchronization problems.

To make further progress on analysis of what one might call networked dynamical systems (which Chazelle calls influences systems), one needs to combine mathematics of algorithms, complexity, combinatorics, and graphs with systems theory and dynamical systems.

Summary and Future Directions

This article presented a brief summary of the literature on flocking and distributed motion co-ordination. Flocking is the process by which various species exhibit synchronous collective motion from simple local interaction rules. Motivated by social aggregation in various species, various algorithms have been developed in the literature to design distributed control laws for group behavior in collective robotics and analysis of opinion dynamics in social networks. The models describe each agent as a kinematic or point mass particle that aligns each agent's direction with that of its neighbors using repeated local averaging of directions. Since the neighborhood structures change due to motion, this results in a distributed switched dynamical system. If a weak notion of connectivity among agents is preserved over time, then agents reach consensus in their direction of motion. Despite the flurry of results in this area, the analysis of this phenomenon that accounts for endogenous change in dynamics is for the most part open.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Oscillator Synchronization

Bibliography

Chatterjee S, Seneta E (1977) Towards consensus: some convergence theorems on repeated averaging. J Appl Probab 14:89–97

Chazelle B (2012) Natural algorithms and influence systems. Commun ACM 55(12):101–110

Cortés J, Martínez S, Bullo F (2006) Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Trans Autom Control 51(8):1289–1298

Cucker F, Smale S (2007) Emergent behavior in flocks. IEEE Trans Autom Control 52(5):852–862

DeGroot MH (1974) Reaching a consensus. J Am Stat Assoc 69(345):118–121

Golub B, Jackson MO (2010) Naive learning in social networks and the wisdom of crowds. Am Econ J Microecon 2(1):112–149

Hegselmann R, Krause U (2002) Opinion dynamics and bounded confidence models, analysis, and simulation. J Artif Soc Soc Simul 5(3):2

Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Autom Control 48(6):988–1001

Moreau L (2005) Stability of multiagent systems with time-dependent communication links. IEEE Trans Autom Control 50(2):169–182

Nedic A, Ozdaglar A (2009) Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Control 54(1):48–61

Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

Reynolds CW (1987) Flocks, herds, and schools: a distributed behavioral model. Comput Graph 21(4):25–34. (SIGGRAPH '87 Conference Proceedings)

Tanner HG, Jadbabaie A, Pappas GJ (2003) Stable flocking of mobile agents, Part I: fixed Topology, Part II: switching topology. In: Proceedings of the 42nd IEEE conference on decision and control, Maui, vol 2. IEEE, pp 2010–2015

Tanner HG, Jadbabaie A, Pappas GJ (2007) Flocking in fixed and switching networks. IEEE Trans Autom Control 52(5):863–868

Tsitsiklis JN (1984) Problems in decentralized decision making and computation (No. LIDS-TH-1424). Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

Tsitsiklis J, Bertsekas D, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE Trans Autom Control 31(9):803–812

Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O (1995) Novel type of phase transition in a system of self-driven particles. Phys Rev Lett 75(6):1226

Force Control in Robotics

Luigi Villani
Dipartimento di Ingeneria Elettrica e Tecnologie
dell'Informazione, Università degli Studi di
Napoli Federico II, Napoli, Italy

Abstract

Force control is used to handle the physical inter-
action between a robot and the environment and
also to ensure safe and dependable operation in
the presence of humans. The control goal may be
that to keep the interaction forces limited or that
to guarantee a desired force along the directions
where interaction occurs while a desired motion
is ensured in the other directions. This entry
presents the basic control schemes, focusing on
robot manipulators.

Keywords

Compliance control; Constrained motion;
Force control; Force/torque sensor; Hybrid
force/motion control; Impedance control;
Stiffness control

Introduction

Control of the physical interaction between
a robot manipulator and the environment is
crucial for the successful execution of a number
of practical tasks where the robot end effector
has to manipulate an object or perform some
operation on a surface. Typical examples in
industrial settings include polishing, deburring,
machining, or assembly.

During contact, the environment may set con-
straints on the geometric paths that can be fol-
lowed by the robot's end effector (kinematic
constraints) as in the case of sliding on a rigid
surface. In other situations, the interaction occurs
with a dynamic environment as in the case of

collaboration with a human. In all cases, a pure
motion control strategy is not recommended, es-
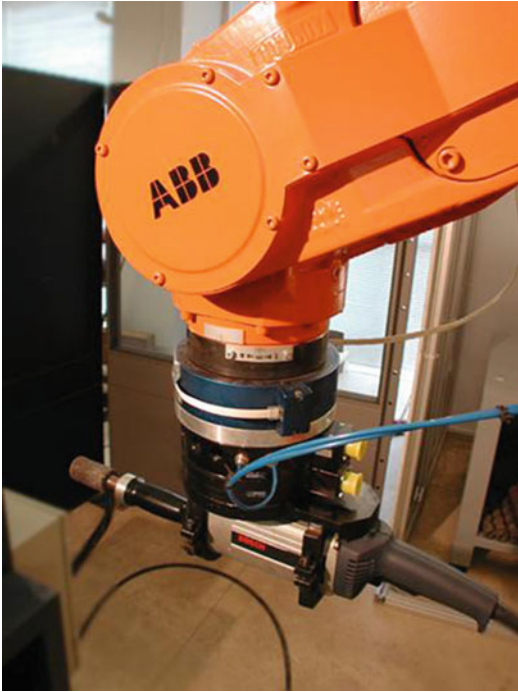pecially if the environment is stiff.

The higher the environment stiffness and po-
sition control accuracy are, the more easily the
contact forces may rise and reach unsafe values.
This drawback can be overcome by introducing
compliance, either in a passive or in an active
fashion, to accommodate the robot motion in
response to interaction forces.

Passive compliance may be due to the struc-
tural compliance of the links, joints, and end ef-
fector or to the compliance of the position servo.
Soft robot arms with elastic joints or links are
purposely designed for intrinsically safe interac-
tion with humans. In contrast, active compliance
is entrusted to the control system, denoted *inter-
action control* or *force control*. In same cases, the
measurement of the contact force and moment
is required, which is fed back to the controller
and used to modify or even generate online the
desired motion of the robot (Whitney 1977).

The passive solution is faster than active re-
action commanded by a computer control algo-
rithm. However, the use of passive compliance
alone lacks of flexibility and cannot guarantee
that high contact forces will never occur. Hence,
the most effective solution is that of using ac-
tive force control (with or without force feed-
back) in combination with some degree of passive
compliance.

In general, six force components are required
to provide complete contact force information:
three translational force components and three
torques. Often, a *force/torque sensor* is mounted
at the robot wrist (see an example in Fig. 1),
but other possibilities exist, for example, force
sensors can be placed on the fingertips of robotic
hands; also, external forces and moments can be
estimated via shaft torque measurements of joint
torque sensors.

The force control strategies can be grouped
into two categories (Siciliano and Villani 1999):
those performing indirect force control and those
performing direct force control. The main dif-
ference between the two categories is that the
former achieve force control via motion control,

$$\mathbf{v}_e = \mathbf{J}(\boldsymbol{q})\dot{\boldsymbol{q}} .$$

The matrix $\mathbf{J}$ is the end-effector Jacobian. For simplicity, the case of nonredundant nonsingular manipulators is considered; therefore, the Jacobian is a square nonsingular matrix.

The force $\boldsymbol{f}_e$ and moment $\boldsymbol{m}_e$ applied by the end effector to the environment are the components of the wrench $\boldsymbol{h}_e = (\boldsymbol{f}_e^T \quad \boldsymbol{m}_e^T)^T$. The joint torques $\boldsymbol{\tau}$ corresponding to $\boldsymbol{h}_e$ can be computed as

$$\boldsymbol{\tau} = \mathbf{J}^T(\boldsymbol{q})\boldsymbol{h}_e .$$

It is useful to consider the operational space formulation of the dynamic model of a rigid robot manipulator in contact with the environment (Khatib 1987):

$$\boldsymbol{\Lambda}(\boldsymbol{q})\dot{\mathbf{v}}_e + \boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})\mathbf{v}_e + \boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{h}_c - \boldsymbol{h}_e , \quad (1)$$

where $\boldsymbol{\Lambda}(\boldsymbol{q})$ is the 6×6 operational space inertia matrix, $\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the wrench including centrifugal and Coriolis effects, and $\boldsymbol{\eta}(\boldsymbol{q})$ is the wrench of the gravitational effects. The vector $\boldsymbol{h}_c = \mathbf{J}^{-T}\boldsymbol{\tau}$ is the equivalent end-effector wrench corresponding to the input joint torques $\boldsymbol{\tau}_c$.

Equation (1) can be seen as a representation of the Newton's Second Law of Motion where all the generalized forces acting on the joints of the robot are reported at the end effector.

The full specification of the system dynamics would require also the analytic description of the interaction force and moment $\boldsymbol{h}_e$. This is a very demanding task from a modeling viewpoint.

The design of the interaction control and the performance analysis are usually carried out under simplifying assumptions. The following two cases are considered:

1. The robot is perfectly rigid, all the compliance in the system is localized in the environment, and the contact wrench is approximated by a linear elastic model.
2. The robot and the environment are perfectly rigid and purely kinematics constraints are imposed by the environment.

It is obvious that these situations are only ideal. However, the robustness of the control should be able to cope with situations where some of the

Force Control in Robotics, Fig. 1 Industrial robot with wrist force/torque sensor and deburring tool

without explicit closure of a force feedback loop; the latter instead offer the possibility of controlling the contact force and moment to a desired value, thanks to the closure of a force feedback loop.

Modeling

The case of interaction of the end effector of a robot manipulator with the environment is considered, which is the most common situation in industrial applications.

The end-effector pose can be represented by the position vector $\boldsymbol{p}_e$ and the rotation matrix $\mathbf{R}_e$, corresponding to the position and orientation of a frame attached to the end effector with respect to a fixed-base frame.

The end-effector velocity is denoted by the 6×1 twist vector $\mathbf{v}_e = (\dot{\boldsymbol{p}}_e^T \quad \boldsymbol{\omega}_e^T)^T$ where $\dot{\boldsymbol{p}}_e$ is the translational velocity and $\boldsymbol{\omega}_e$ is the angular velocity and can be computed from the joint velocity vector $\dot{\boldsymbol{q}}$ using the linear mapping

ideal assumptions are relaxed. In that case the control laws may be adapted to deal with nonideal characteristics.

Indirect Force Control

The aim of indirect force control is that of achieving a desired compliant dynamic behavior of the robot's end effector in the presence of interaction with the environment.

Stiffness Control

The simpler approach is that of imposing a suitable static relationship between the deviation of the end-effector position and orientation from a desired pose and the force exerted on the environment, by using the control law

$$\mathbf{h}_c = \mathbf{K}_P \Delta \mathbf{x}_{de} - \mathbf{K}_D \mathbf{v}_e + \boldsymbol{\eta}(\mathbf{q}) , \qquad (2)$$

where $\mathbf{K}_P$ and $\mathbf{K}_D$ are suitable matrix gains and $\Delta \mathbf{x}_{de}$ is a suitable error between a desired and the actual end-effector position and orientation. The position error component of $\Delta \mathbf{x}_{de}$ can be simply chosen as $\mathbf{p}_d - \mathbf{p}_e$. Concerning the orientation error component, different choices are possible (Caccavale et al. 1999), which are not all equivalent, but this issue is outside the scope of this entry.

The control input (2) corresponds to a wrench (force and moment) applied to the end effector, which includes a gravity compensation term $\boldsymbol{\eta}(\mathbf{q})$, a viscous damping term $\mathbf{K}_D \mathbf{v}_e$, and an elastic wrench provided by a virtual spring with stiffness matrix $\mathbf{K}_P$ (or, equivalently, compliance matrix $\mathbf{K}_P^{-1}$) connecting the end-effector frame with a frame of desired position and orientation. This control law is known as *stiffness control* or *compliance control* (Salisbury 1980).

Using the Lyapunov method, it is possible to prove the asymptotic stability of the equilibrium solution of equation

$$\mathbf{K}_P \Delta \mathbf{x}_{de} = \mathbf{h}_e ,$$

meaning that, at steady state, the robot's end effector has a desired elastic behavior under the action of the external wrench $\mathbf{h}_e$. It is clear that, if $\mathbf{h}_e \neq \mathbf{0}$, then the end effector deviates from the desired pose, which is usually denoted as *virtual pose*.

Physically, the closed-loop system (1) with (2) can be seen as a 6-DOF nonlinear and configuration-dependent mass-spring-damper system with inertia (mass) matrix $\boldsymbol{\Lambda}(\mathbf{q})$ and adjustable damping $\mathbf{K}_D$ and stiffness $\mathbf{K}_P$, under the action of the external wrench $\mathbf{h}_e$.

Impedance Control

A configuration-independent dynamic behavior can be achieved if the measure of the end-effector force and moment $\mathbf{h}_e$ is available, by using the control law, known as *impedance control* (Hogan 1985):

$$\mathbf{h}_c = \boldsymbol{\Lambda}(\mathbf{q})\boldsymbol{\alpha} + \boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\eta}(\mathbf{q}) + \mathbf{h}_e ,$$

where $\boldsymbol{\alpha}$ is chosen as:

$$\boldsymbol{\alpha} = \dot{\mathbf{v}}_d + \mathbf{K}_M^{-1}(\mathbf{K}_D \Delta \mathbf{v}_{de} + \mathbf{K}_P \Delta \mathbf{x}_{de} - \mathbf{h}_e) .$$

The following expression can be found for the closed-loop system

$$\mathbf{K}_M \Delta \dot{\mathbf{v}}_{de} + \mathbf{K}_D \Delta \mathbf{v}_{de} + \mathbf{K}_P \Delta \mathbf{x}_{de} = \mathbf{h}_e , \qquad (3)$$

representing the equation of a 6-DOF configuration-independent mass-spring-damper system with adjustable inertia (mass) matrix $\mathbf{K}_M$, damping $\mathbf{K}_D$, and stiffness $\mathbf{K}_P$, known as mechanical *impedance*.

A block diagram of the resulting impedance control is sketched in Fig. 2.

The selection of good impedance parameters ensuring a satisfactory behavior is not an easy task and can be simplified under the hypothesis that all the matrices are diagonal, resulting in a decoupled behavior for the end-effector coordinates.

Moreover, the dynamics of the controlled system during the interaction depends on the dynamics of the environment that, for simplicity, can be approximated as a simple elastic law for each coordinate, of the form
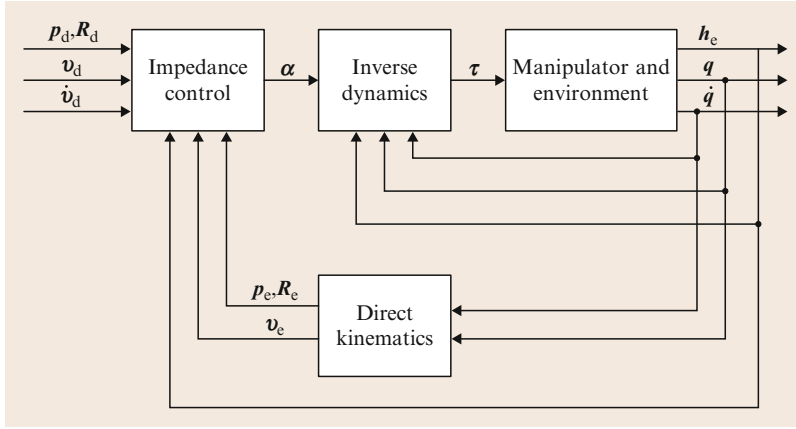
Force Control in Robotics, Fig. 2 Impedance control

$$h_e = k\Delta x_{eo} \; ,$$

where $\Delta x_{eo} = x_e - x_o$, while x_o and k are the undeformed position and the stiffness coefficient of the spring, respectively.

In the above hypotheses, the transient behavior of each component of Eq. (3) can be set by assigning the natural frequency and damping ratio with the relations

$$\omega_n = \sqrt{\frac{k_P + k}{k_M}} \; , \quad \zeta = \frac{1}{2}\frac{k_D}{\sqrt{k_M(k_P + k)}} \; .$$

Hence, if the gains are chosen so that a given natural frequency and damping ratio are ensured during the interaction (i.e., for $k \neq 0$), a smaller natural frequency with a higher damping ratio will be obtained when the end effector moves in free space (i.e., for $k = 0$). As for the steady-state performance, the end-effector error and the interaction force for the generic component are

$$\Delta x_{de} = \frac{k}{(k_P + k)}\Delta x_{do} \; , \quad h = \frac{k_P k}{k_P + k}\Delta x_{do} \; ,$$

showing that, during interaction, the contact force can be made small at the expense of a large position error in steady state, as long as the robot stiffness k_P is set low with respect

to the stiffness of the environment k and vice versa.

Direct Force Control

Indirect force control does not require explicit knowledge of the environment, although to achieve a satisfactory dynamic behavior, the control parameters have to be tuned for a particular task. On the other hand, a model of the interaction task is usually required for the synthesis of direct force control algorithms.

In the following, it is assumed that the environment is rigid and frictionless and imposes kinematic constraints to the robot's end-effector motion (Mason 1981). These constraints reduce the dimension of the space of the feasible end-effector velocities and of the contact forces and moments. In detail, in the presence of m independent constraints ($m < 6$), the end-effector velocity belongs to a subspace of dimension $6 - m$, while the end-effector wrench belongs to a subspace of dimension m and can be expressed in the form

$$\mathbf{v}_e = \mathbf{S}_v(\mathbf{q})\mathbf{\nu} \; , \quad \mathbf{h}_e = \mathbf{S}_f(\mathbf{q})\mathbf{\lambda}$$

where $\mathbf{\nu}$ is a suitable $(6 - m) \times 1$ vector and $\mathbf{\lambda}$ is a suitable $m \times 1$ vector. Moreover, the subspaces of forces and velocity are *reciprocal*, i.e.:

$$\boldsymbol{h}_{\mathrm{e}}^{T}\mathbf{v}_{\mathrm{e}} = 0\,, \quad \mathbf{S}_{\mathrm{f}}^{T}(\boldsymbol{q})\mathbf{S}_{\mathrm{v}}(\boldsymbol{q}) = \mathbf{0}\,.$$

The concept of reciprocity expresses the physical fact that, in the hypothesis of rigid and frictionless contact, the wrench does not cause any work against the twist.

An interaction task can be assigned in terms of a desired end-effector twist $\mathbf{v}_{\mathrm{d}}$ and wrench $\boldsymbol{h}_{\mathrm{d}}$ that are computed as:

$$\mathbf{v}_{\mathrm{d}} = \mathbf{S}_{\mathrm{v}}\boldsymbol{v}_{\mathrm{d}}\,, \quad \boldsymbol{h}_{\mathrm{d}} = \mathbf{S}_{\mathrm{f}}\boldsymbol{\lambda}_{\mathrm{d}}\,,$$

by specifying vectors $\boldsymbol{\lambda}_{\mathrm{d}}$ and $\boldsymbol{v}_{\mathrm{d}}$.

In many robotic tasks it is possible to set an orthogonal reference frame, usually referred as *task frame* (De Schutter and Van Brussel 1988), in which the matrices $\mathbf{S}_{\mathrm{v}}$ and $\mathbf{S}_{\mathrm{f}}$ are constant. Moreover, the interaction task is specified by assigning a desired force/torque or a desired linear/angular velocity along/about each of the frame axes.

An example of task frame definition and task specification is given below.

Peg-in-Hole: The goal of this task is to push the peg into the hole while avoiding wedging and jamming. The peg has two degrees of motion freedom; hence, the dimension of the velocity-controlled subspace is $6 - m = 2$, while the dimension of the force-controlled subspace is $m = 4$. The task frame can be chosen as shown in Fig. 3, and the task can be achieved by assigning the following desired forces and torques:

- Zero forces along the x_{t} and y_{t} axes
- Zero torques about the x_{t} and y_{t} axes and the desired velocities
- A nonzero linear velocity along the z_{t}-axis
- An arbitrary angular velocity about the z_{t}-axis

The task continues until a large reaction force in the z_{t} direction is measured, indicating that the peg has hit the bottom of the hole, not represented in the figure. Hence, the matrices $\mathbf{S}_{\mathrm{f}}$ and $\mathbf{S}_{\mathrm{v}}$ can be chosen as

$$\mathbf{S}_{\mathrm{f}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}\,, \quad \mathbf{S}_{\mathrm{v}} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}\,.$$
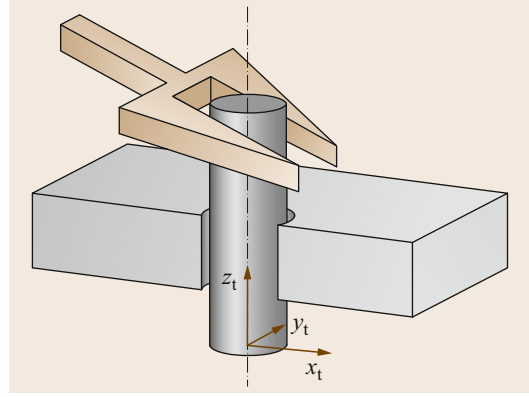
Force Control in Robotics, Fig. 3 Insertion of a cylindrical peg into a hole

The task frame can be chosen attached either to the end effector or to the environment.

Hybrid Force/Motion Control

The reciprocity of the velocity and force subspaces naturally leads to a control approach, known as *hybrid force/motion control* (Raibert and Craig 1981; Yoshikawa 1987), aimed at controlling simultaneously both the contact force and the end-effector motion in two reciprocal subspaces.

The reduced order dynamics of the robot with kinematic constraints is described by $6 - m$ second-order equations

$$\boldsymbol{\Lambda}_{\mathrm{v}}(\boldsymbol{q})\dot{\boldsymbol{v}} = \mathbf{S}_{\mathrm{v}}^{T}\big[\boldsymbol{h}_{\mathrm{c}} - \boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}})\big]\,,$$

where $\boldsymbol{\Lambda}_{\mathrm{v}} = \mathbf{S}_{\mathrm{v}}^{T}\boldsymbol{\Lambda}\mathbf{S}_{\mathrm{v}}$ and $\boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\mathbf{v}_{\mathrm{e}} + \boldsymbol{\eta}(\boldsymbol{q})$, assuming constant matrices $\mathbf{S}_{\mathrm{v}}$ and $\mathbf{S}_{\mathrm{f}}$. Moreover, the vector $\boldsymbol{\lambda}$ can be computed as

$$\boldsymbol{\lambda} = \mathbf{S}_{\mathrm{f}}^{\dagger}(\boldsymbol{q})\big[\boldsymbol{h}_{\mathrm{c}} - \boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}})\big]\,,$$

revealing that the contact force is a constraint force which instantaneously depends on the applied input wrench $\boldsymbol{h}_{\mathrm{c}}$.

An inverse-dynamics inner control loop can be designed by choosing the control wrench $\boldsymbol{h}_{\mathrm{c}}$ as

$$\boldsymbol{h}_{\mathrm{c}} = \boldsymbol{\Lambda}(\boldsymbol{q})\mathbf{S}_{\mathrm{v}}\boldsymbol{\alpha}_{\mathrm{v}} + \mathbf{S}_{\mathrm{f}}\boldsymbol{f}_{\lambda} + \boldsymbol{\mu}(\boldsymbol{q},\dot{\boldsymbol{q}})\,,$$

where $\boldsymbol{\alpha}_v$ and $\boldsymbol{f}_\lambda$ are properly designed control inputs, which leads to the equations

$$\dot{\boldsymbol{v}} = \boldsymbol{\alpha}_v, \quad \boldsymbol{\lambda} = \boldsymbol{f}_\lambda \ ,$$

showing a complete decoupling between motion control and force control.

Then, the desired force $\boldsymbol{\lambda}_d(t)$ can be achieved by setting

$$\boldsymbol{f}_\lambda = \boldsymbol{\lambda}_d(t) \ ,$$

but this choice is very sensitive to disturbance forces, since it contains no force feedback. Alternative choices are

$$\boldsymbol{f}_\lambda = \boldsymbol{\lambda}_d(t) + \mathbf{K}_{P\lambda}\big[\boldsymbol{\lambda}_d(t) - \boldsymbol{\lambda}(t)\big] \ ,$$

or

$$\boldsymbol{f}_\lambda = \boldsymbol{\lambda}_d(t) + \mathbf{K}_{I\lambda} \int_0^t \big[\boldsymbol{\lambda}_d(\tau) - \boldsymbol{\lambda}(\tau)\big]\mathrm{d}\tau \ ,$$

where $\mathbf{K}_{P\lambda}$ and $\mathbf{K}_{I\lambda}$ are suitable positive-definite matrix gains. The proportional feedback is able to reduce the force error due to disturbance forces, while the integral action is able to compensate for constant bias disturbances.

Velocity control is achieved by setting

$$\boldsymbol{\alpha}_v = \dot{\boldsymbol{v}}_d(t) + \mathbf{K}_{Pv}\big[\boldsymbol{v}_d(t) - \boldsymbol{v}(t)\big]$$
$$+ \mathbf{K}_{Iv} \int_0^t \big[\boldsymbol{v}_d(\tau) - \boldsymbol{v}(\tau)\big]\mathrm{d}\tau \ ,$$

where $\mathbf{K}_{Pv}$ and $\mathbf{K}_{Iv}$ are suitable matrix gains. It is straightforward to show that asymptotic tracking of $\boldsymbol{v}_d(t)$ and $\dot{\boldsymbol{v}}_d(t)$ is ensured with exponential convergence for any choice of positive-definite matrices $\mathbf{K}_{Pv}$ and $\mathbf{K}_{Iv}$.

Notice that the implementation of force feedback requires the computation of vector $\boldsymbol{\lambda}$ from the measurement of the end-effector wrench $\boldsymbol{h}_e$ as $\mathbf{S}_f^\dagger \mathbf{v}_e$, being $\mathbf{S}_f^\dagger$ a suitable pseudoinverse of matrix $\mathbf{S}_f$. Analogously, vector $\boldsymbol{v}$ can be computed from $\mathbf{v}_e$ as $\mathbf{S}_v^\dagger \mathbf{v}_e$.

The hypothesis of rigid contact can be removed, and this implies that along some directions both motion and force are allowed, although they are not independent. Hybrid force/motion control schemes can be defined also in this case Villani and De Schutter (2008).

Summary and Future Directions

This entry has sketched the main approaches to force control in a unifying perspective. However, there are many aspects that have not been considered here. The two major paradigms of force control (impedance and hybrid force/motion control) are based on several simplifying assumptions that are only partially satisfied in practical implementations and that have been partially removed in more advanced control methods.

Notice that the performance of a force-controlled robotic system depends on the interaction with a changing environment which is very difficult to model and identify correctly. Hence, the standard performance indices used to evaluate a control system, i.e., stability, bandwidth, accuracy, and robustness, cannot be defined by considering the robotic system alone, as for the case of robot motion control, but must be always referred to the particular contact situation at hand.

Force control in industrial applications can be considered as a mature technology, although, for the reason explained above, standard design methodologies are not yet available. Force control techniques are employed also in medical robotics, haptic systems, telerobotics, humanoid robotics, micro-robotics, and nano robotics. An interesting field of application is related to human-centered robotics, where control plays a key role to achieve adaptability, reaction capability, and safety. Robots and biomechatronic systems based on the novel variable impedance actuators, with physically adjustable compliance and damping, capable to react softly when touching the environment, necessitate the design of specific control laws. The combined use of exteroceptive sensing (visual, depth, proximity, force, tactile sensing) for reactive control in the presence of uncertainty represents another challenging research direction.

Cross-References

► Robot Grasp Control
► Robot Motion Control

Recommended Reading

This entry has presented a brief overview of the basic force control techniques, and the cited references represent a selection of the main pioneering contributions. A more extensive treatment of this topic with related bibliography can be found in Villani and De Schutter (2008). Besides impedance control and hybrid force/position control, an approach designed to cope with uncertainties in the environment geometry is the parallel force/position control (Chiaverini and Sciavicco 1993; Chiaverini et al. 1994). In the paper Ott et al. (2008) the passive compliance of lightweight robots is combined with the active compliance ensured by impedance control. A systematic constraint-based methodology to specify complex tasks has been presented by De Schutter et al. (2007).

Bibliography

Caccavale F, Natale C, Siciliano B, Villani L (1999) Six-DOF impedance control based on angle/axis representations. IEEE Trans Robot Autom 15:289–300

Chiaverini S, Sciavicco L (1993) The parallel approach to force/position control of robotic manipulators, IEEE Trans Robot Autom 9:361–373

Chiaverini S, Siciliano B, Villani L (1994) Force/position regulation of compliant robot manipulators. IEEE Trans Autom Control 39:647–652

De Schutter J, Van Brussel H (1988) Compliant robot motion I. A formalism for specifying compliant motion tasks. Int J Robot Res 7(4):3–17

De Schutter J, De Laet T, Rutgeerts J, Decré W, Smits R, Aerbeliën E, Claes K, Bruyninckx H (2007) Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. Int J Robot Res 26(5):433–455

Hogan N (1985) Impedance control: an approach to manipulation: parts I–III. ASME J Dyn Syst Meas Control 107:1–24

Khatib O (1987) A unified approach for motion and force control of robot manipulators: the operational space formulation. IEEE J Robot Autom 3:43–53

Mason MT(1981) Compliance and force control for computer controlled manipulators. IEEE Trans Syst Man Cybern 11:418–432

Ott C, Albu-Schaeffer A, Kugi A, Hirzinger G (2008) On the passivity based impedance control of flexible joint robots. IEEE Trans Robot 24:416–429

Raibert MH, Craig JJ (1981) Hybrid position/force control of manipulators. ASME J Dyn Syst Meas Control 103:126–133

Salisbury JK (1980) Active stiffness control of a manipulator in Cartesian coordinates. In: 19th IEEE conference on decision and control, Albuquerque, pp 95–100

Siciliano B, Villani L (1999) Robot force control. Kluwer, Boston

Villani L, De Schutter J (2008) Robot force control. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 161–185

Whitney DE (1977) Force feedback control of manipulator fine motions. ASME J Dyn Syst Meas Control 99:91–97

Yoshikawa T (1987) Dynamic hybrid position/force control of robot manipulators – description of hand constraints and calculation of joint driving force. IEEE J Robot Autom 3:386–392

Frequency Domain System Identification

Johan Schoukens and Rik Pintelon
Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Abstract

In this chapter we give an introduction to frequency domain system identification. We start from the identification work loop in ► System Identification: An Overview, Fig. 4, and we discuss the impact of selecting the time or frequency domain approach on each of the choices that are in this loop. Although there is a full theoretical equivalence between the time and frequency domain identification approach, it turns out that, from practical point of view, there can be a natural preference for one of both domains.

Keywords

Discrete and continuous time models; Experiment setup; Frequency and time domain identification; Plant and noise model

Introduction

System identification provides methods to build a mathematical model for a dynamical system starting from measured input and output signals (▶ System Identification: An Overview; see the section "Models and System Identification"). Initially, the field was completely dominated by the time domain approach, and the frequency domain was used to interpret the results (Ljung and Glover 1981). This picture changed in the nineteenth of the last century by the development of advanced frequency domain methods (Ljung 2006; Pintelon and Schoukens 2012), and nowadays it is widely accepted that there is a full theoretical equivalence between time and frequency domain system identification under some weak conditions (Agüero et al. 2009; Pintelon and Schoukens 2012). Dedicated toolboxes are available for both domains (Kollar 1994; Ljung 1988). This raises the question how to choose between time and frequency domain identification. Many times the choice between both approaches can be made based on the user's familiarity with one of two methods. However, for some problems it turns out that there is a natural preference for the time or the frequency domain. This contribution discusses the main issues that need to be considered when making this choice, and it provides additional insights to guide the reader to the best solutions for her/his problem.

In the identification work loop ▶ System Identification: An Overview, Fig. 4, we need to address three important questions (Ljung 1999, Sect. 1.4; Söderström and Stoica 1989, Chap. 1; Pintelon and Schoukens 2012, Sect. 1.4) that directly interact with the choice between time and frequency domain system identification:

- What data are available? What data are needed? This discussion will influence the selection of the measurement setup, the model choice, and the design of the experiment.
- What kind of models will be used? We will mainly focus on the identification of discrete time and continuous time models, using exactly the same frequency domain tools.

- How will the model be matched to the data? This question boils down to the choice of a cost function that measures the distance between the data and the model. We will discuss the use of nonparametric weighting functions in the frequency domain.

In the next sections, we will address these and similar questions in more detail. First, we discuss the measurement of the raw data. The choices that are made in this step will have a strong impact on many user aspects of the identification process. The frequency domain formulation will turn out to be a natural choice to propose a unified formulation of the system identification problem, including discrete and continuous time modeling. Next, a generalized frequency domain description of the system relation will be proposed. This model will be matched to the data, using a weighted least squares cost function, formulated in the frequency domain identification. This will allow for the use of nonparametric weighting functions, based on a nonparametric preprocessing of the data. Eventually, some remaining user aspects are discussed.

Data Collection

In this section, we discuss the measurement assumptions that are made when the raw data are collected. It will turn out that these will directly influence the natural choice of the models that are used to describe the continuous time physical systems.

### Time Domain and Frequency Domain Measurements

The data can be collected either in the time or in the frequency domain, and we discuss briefly both options.

Time Domain Measurements

Most measurements are nowadays made in the time domain because very fast high-quality analog-to-digital convertors (ADC) became available at a low price. These allow us to sample

Frequency Domain System Identification, Fig. 1 Comparison of the ZOH and BL signal reconstruction of a discrete time sequence: (**a**) time domain: $\cdots$ BL, – ZOH; (**b**) spectrum ZOH signal; (**c**) spectrum BL signal
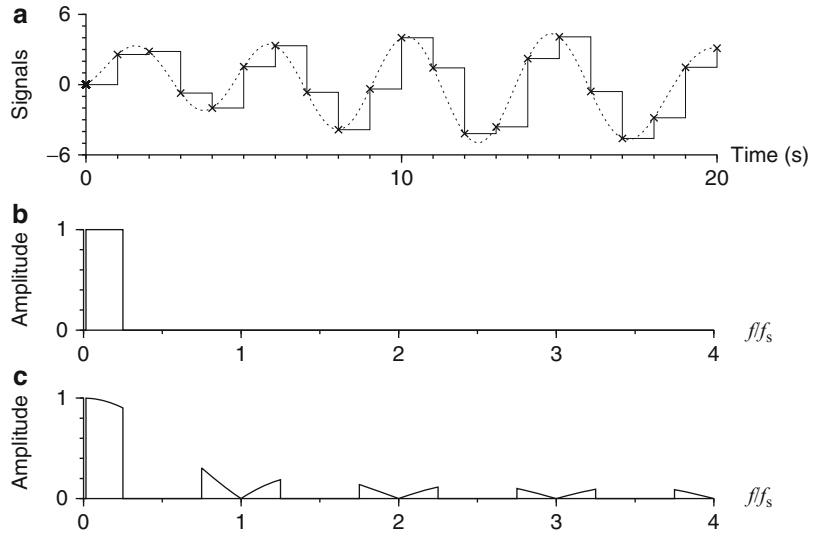
and discretize the continuous time input and output signals and process these on a digital computer. Also the excitation signals are mostly generated from a discrete time sequence with a digital-to-analog convertor (DAC).

The continuous time input and output signals $u_c(t), y_c(t)$ of the system to be modeled are measured at the sampling moments $t_k = kT_s$, with $T_s = 1/f_s$ the sampling period and f_s the sampling frequency: $u(k) = u_c(kT_s)$, and $y(k) = y_c(kT_s)$. The discrete time signals $u(k), y(k), k = 1, \cdots, N$ are transformed to the frequency domain using the discrete Fourier transform (DFT) (▶ Nonparametric Techniques in System Identification, Eq. 1), resulting in the DFT spectra $U(l), Y(l)$, at the frequencies $f_l = l\frac{f_s}{N}$. Making some abuse of notation, we will reuse the same symbols later in this text, to denote the Z-transform of the signals, for example, $Y(z)$ will also be used for the Z-transform of $y(k)$.

Frequency Domain Measurements

A major exception to this general trend toward time domain measurements are the (high-frequency) network analyzers that measure the transfer function of a system frequency by frequency, starting from the steady-state response to a sine excitation. The frequency is stepped over the frequency

band of interest, resulting directly in a measured frequency response function at a user selected set of frequencies $\omega_k, k = 1, \cdots, F$:

$$G(\omega_k).$$

From the identification point of view, we can easily fit the latter situation in the frequency domain identification framework, by putting

$$U(k) = 1, Y(k) = G(\omega_k).$$

For that reason we will focus completely on the time domain measurement approach in the remaining part of this contribution.

Zero-Order-Hold and Band-Limited Setup: Impact on the Model Choice

No information is available on how the continuous time signals $u_c(t), y_c(t)$ vary in between the measured samples $u(k), y(k)$. For that reason we need to make an assumption and make sure that the measurement setup is selected such that the intersample assumption is met. Two intersample assumptions are very popular (Pintelon and Schoukens 2012; Schoukens et al. 1994, pp. 498–512): the zero-order-hold (ZOH) and the band-limited assumption (BL). Both options are shown in Fig. 1 and discussed below. The choice

of these assumptions does not only affect the measurement setup; it has also a strong impact on the selection between a continuous or a discrete time model choice.

Zero-Order Hold

The ZOH setup assumes that the excitation remains constant in between the samples. In practice, the model is identified between the discrete time reference signal in the memory of the generator and the sampled output. The intersample behavior is an intrinsic part of the model: if the intersample behavior changes, also the corresponding model will change. The ZOH assumption is very popular in digital control. In that case the sampling frequency f_s is commonly chosen 10 times larger than the frequency band of interest.

A discrete time model gives, in case of noise-free data, an exact description between the sampled input $u(k)$ and output $y(k)$ of the continuous time system:

$$y(k) = G(q, \theta)u(k)$$

in the time domain (▶ System Identification: An Overview, Eq. 6). In this expression, q denotes the shift operator (time domain), and it is replaced by z in the z-domain description (transfer function description):

$$Y(z) = G(z, \theta)U(z).$$

Evaluating the transfer function at the unit circle by replacing $z = e^{i\omega}$ results in the frequency domain description of the system:

$$Y(e^{i\omega}) = G(e^{i\omega}, \theta)U(e^{i\omega})$$

(▶ System Identification: An Overview, Eq. 34).

Band-Limited Setup

The BL setup assumes that above a given frequency $f_{max} < f_s/2$, there is no power in the signals. The continuous time signals are filtered by well-tuned anti-alias filters (cutoff frequency $f_{max} < f_s/2$), before they are sampled. Outside the digital control world, the BL setup is the standard choice for discrete time measurements. Without using anti-alias filters, large errors can be created due to aliasing effects: the high frequency ($f > f_s/2$) content of the measured signals is folded down in the frequency band of interest and act there as a disturbance. For that reason it is strongly advised to use always anti-alias filters in the measurement setup.

The exact relation between BL signals is described by a continuous time model, for example, in the frequency domain:

$$Y(\omega) = G(\omega, \theta)U(\omega)$$

(Schoukens et al. 1994).

Combining Discrete Time Models and BL Data

It is also possible to identify a discrete time model between the BL data, at a cost of creating (very) small model errors (Schoukens et al. 1994). This is the standard setup that is used in digital signal procession applications like digital audio processing. The level of the model errors can be reduced by lowering the ratio f_{max}/f_s or by increasing the model order. In a properly designed setup, the discrete time model errors can be made very small, e.g., relative errors below 10^{-5}. In Table 1 an overview of the models corresponding to the experimental conditions is given.

Extracting Continuous Time Models from ZOH Data

Although the most robust and practical choice to identify a continuous time model is to start from BL data, it is also possible to extract a continuous time model under the ZOH setup. A first possibility is to assume that the ZOH assumption is perfectly met, which is a very hard assumption to realize in practice. In that case the continuous time model can be retrieved by a linear step invariant transformation of the discrete time model. A second possibility is to select a very high sample frequency with respect to the bandwidth of the system. In that case it is advantageous to describe the discrete time model using the delta operator (Goodwin 2010), and

Frequency Domain System Identification, Table 1 Relations between the continuous time system $G(s)$ and the identified models as a function of the signal and model choices

	DT-model (Assuming ZOH-setup)	CT-model (Assuming BL-setup)
ZOH setup	Exact DT-model $$G(z) = \left(1 - z^{-1}\right) Z \left\{\frac{G(s)}{S}\right\}$$ **'standard conditions DT modelling'**	Not studied
BL setup	Approximate DT model $$\tilde{G}(z)\, \tilde{G}\left(z = e^{j\omega T_s}\right) \approx G(s = j\omega),\ |\omega| < \frac{\omega_s}{2}$$ **'digital signal processing field'**	Exact CT-model $G(s)$ **'standard conditions CT modelling'**

we have that the coefficients of the discrete time model converge to those of the continuous time model.

Models of Cascaded Systems

In some problems we want to build models for a cascade of two systems G_1, G_2. It is well known that the overall transfer function G is given by the product $G(\omega) = G_1(\omega)G_2(\omega)$. This result holds also for models that are identified under the BL signal assumption: the model for the cascade will be the product of the models of the individual systems. However, the result does not hold under the ZOH assumption, because the intermediate signal between G_1, G_2 does not meet the ZOH assumption. For that reason, the ZOH model of a cascaded system is not obtained by cascading the ZOH models of the individual systems.

Experiment Design: Periodic or Random Excitations?

In general, arbitrary data can be used to identify a system as long as some basic requirements are respected (▶ System Identification: An Overview, section on Experiment Design). Imposing periodic excitations can be an important restriction of the user's freedom to design the experiment, but we will show in the next sections that it offers also major advantages at many steps in the identification work loop (▶ System Identification: An Overview, Fig. 4).

With the availability of arbitrary wave form generators, it became possible to generate arbitrary periodic signals. The user should make two major choices during the design of the periodic excitation: the selection of the amplitude spectrum (How is the available power distributed over the frequency?) and the choice of the frequency resolution (What is the frequency step between two successive points of the measured FRF?) (Pintelon and Schoukens 2012, Sect. 5.3).

The amplitude spectrum is mainly set by the requirement that the excited frequency band should cover the frequency band of interest. A white noise excitation covers the full frequency band, including those bands that are of no interest for the user. This is a waste of power and it should be avoided. Designing a good power spectrum for identification and control purposes is discussed in ▶ Experiment Design and Identification for Control.

The frequency resolution $f_0 = 1/T$ is set by the inverse of the period of the signal. It should be small enough so that no important dynamics are missed, e.g., a very sharp mechanical resonance.

The reader should be aware that exactly the same choices have to be made during the design of nonperiodic excitations. If, for example, a random noise excitation is used, the frequency resolution is also restricted by the length of the experiment T_m and the corresponding frequency resolution is again $f_0 = 1/T_m$. The power spectrum of the noise excitation should be well shaped using a digital filter.

Nonparametric Preprocessing of the Data in the Frequency Domain

Before a parametric model is identified from the raw data, a lot of information can be gained, almost for free, by making a nonparametric analysis of the data. This can be done with very

little user interaction. Some of these methods are explicitly linked to the periodic nature of the data, other methods apply also to random excitations.

Nonparametric Frequency Analysis of Periodic Data

By using simple DFT techniques, the following frequency domain information is extracted from sampled time domain data $u(t), y(t), t = 1, \ldots, N$ (Pintelon and Schoukens 2012):

- The signal information: $U(l), Y(l)$, the DFT spectra of the input and output, evaluated at the frequencies $l f_0$, with $k = 1, 2, \cdots, F$.
- Disturbing noise variance information: The full data record is split, so that each subrecord contains a single period. For each of these, the DFT spectrum is calculated. Since the signals are periodic, they do not vary from one period to the other, so that the observed variations can be attributed to the noise. By calculating the sample mean and variance over the periods at each frequency, a nonparametric noise analysis is available. The estimated variances $\hat{\sigma}_U^2(k), \hat{\sigma}_Y^2(k)$ measure the disturbing noise power spectrum at frequency f_k on the input and the output respectively. The covariance $\hat{\sigma}_{YU}^2(k)$ characterizes the linear relations between the noise on the input and the output.

This is very valuable information because, even before starting the parametric identification step, we get already full access to the quality of the raw data. As a consequence, there is also no interference between the plant model estimation and the noise analysis: plant model errors do not affect the estimated noise model. It is also important to realize that there is no user interaction requested to make this analysis and it follows directly from a simple DFT analysis of the raw data. These are two major advantages of using periodic excitations.

Nonlinear Analysis

Using well-designed periodic excitations, it is possible to detect the presence of nonlinear distortions during the nonparametric frequency step. The level of the nonlinear distortions at the output

of the system is measured as a function of the frequency, and it is even possible to differentiate between even (e.g., x^2) and odd (e.g., x^3) distortions. While the first only act as disturbing noise in a linear modeling framework, the latter will also affect the linearized dynamics and can change, for example, the pole positions of a system (Pintelon and Schoukens 2012, Sect. 4.3).

Noise and Data Reduction

By averaging the periodic signals over the successive periods, we get a first reduction of the noise. An additional noise reduction is possible when not all frequencies are excited. If a very wide frequency band has to be covered, a fine frequency resolution is needed at the low frequencies, whereas in the higher frequency bands, the resolution can be reduced. Signals with a logarithmic frequency distribution are used for that purpose. Eliminating the unexcited frequencies does not only reduce the noise, it also reduces significantly the amount of raw data to be processed. By combining different experiments that cover each a specific frequency band, it is possible to measure a system over multiple decades, e.g., electrical machines are measured from a few mHz to a few kHz.

In a similar way, it is also possible to focus the fit on the frequency band of interest by including only those frequencies in the parametric modeling step.

High-Quality Frequency Response Function Measurements

For periodic excitations, it is very simple to obtain high-quality measurements of the nonparametric frequency response function of the system. These results can be extended to random excitations at a cost of using more advanced algorithms that require more computation time (Pintelon and Schoukens, Chap. 7). This approach is discussed in detail in ▶ Nonparametric Techniques in System Identification (Eq. 1).

Generalized Frequency Domain Models

A very important step in the identification work loop is the choice of the model class (▶ System Identification: An Overview, Fig. 4). Although most physical systems are continuous time, the models that we need might be either discrete time (e.g., digital control, computer simulations, digital signal processing) or continuous time (e.g., physical interpretation of the model, analog control) (Ljung 1999, Sects. 2.1 and 4.3). A major advantage of the frequency domain is that both model classes are described by the same transfer function model. The only difference is the choice of the frequency variable. For continuous time models we operate in the Laplace domain, and the frequency variable is retrieved on the imaginary axis by putting $s = j\omega$. For discrete time systems we work on the unit circle that is described by the frequency variable $z = e^{j 2\pi f/f_s}$. The application class can be even extended to include diffusion phenomena by putting $\Omega = \sqrt{j\omega}$ (Pintelon et al. 2005). From now on we will use the generalized frequency variable Ω, and depending on the selected domain, the proper substitution $j\omega$, $e^{j 2\pi f/f_s}$, $\sqrt{j\omega}$ should be made. The unified discrete and continuous time description

$$Y(k) = G(\Omega_k, \theta)U(k).$$

illustrates also very nicely that in the frequency domain there is a strong similarity between discrete and continuous time system identification.

To apply this model to finite length measurements, it should be generalized to include the effect of the initial conditions (time domain) or the begin and end effects (called leakage in the frequency domain). See ▶ Nonparametric Techniques in System Identification, "The Leakage Problem" section. The amazing result is that in the frequency domain, both effects are described by exactly the same mathematical expression. This leads eventually to the following model in the frequency domain that is valid for periodic and arbitrary (nonperiodic) BL or ZOH excitations (Pintelon et al. 1997; McKelvey 2002; Pintelon and Schoukens 2012, Chap. 6):

$$Y(k) = G(\Omega, \theta)U(k) + T_G(\Omega, \theta)$$

which becomes for SISO (single-input-single-output) systems:

$$G(\Omega, \theta) = \frac{B(\Omega, \theta)}{A(\Omega, \theta)}, \text{ and } T_G(\Omega, \theta) = \frac{I(\Omega, \theta)}{A(\Omega, \theta)}.$$

A, B, I are all polynomials in Ω. The transient term $T_G(\Omega, \theta)$ models transient and leakage effects. It is most important for the reader to realize that this is an exact description for noise free data. Observe that it is very similar to the description in the time domain: $y(t) = G(q, \theta)u(t) + t_G(t, \theta)$. In that case the transient term $t_G(t, \theta)$ models the initial transient that is due to the initial conditions.

Parametric Identification

Once we have the data and the model available in the frequency domain, we define the following weighted least squares cost function to match the model to the data (Schoukens et al. 1997):

$$V(\theta) = \frac{1}{F} \sum_{k=1}^{F} \frac{\left| \hat{Y}(k) - G(\Omega_k, \theta) \hat{U}(k) - T_G(\Omega_k, \theta) \right|^2}{\hat{\sigma}_Y^2(k) + \hat{\sigma}_U^2(k) |G(\Omega_k, \theta)|^2 - 2\mathrm{Re}(\hat{\sigma}_{YU}^2(k) G(\Omega_k, \theta))}.$$

The properties of this estimator are fully studied in Schoukens et al. (1999) and Pintelon and Schoukens (2012, Sect. 10.3), and it is shown that it is a consistent and almost efficient estimator under very mild conditions.

The formal link with the time domain cost function, as presented in ▶ System Identification: An Overview, can be made by assuming that the input is exactly known ($\hat{\sigma}_U^2(k) = 0$, $\hat{\sigma}_{YU}^2(k) = 0$), and replacing the nonparametric

noise model on the output by a parametric model:

$$\hat{\sigma}_Y^2(k) = \lambda \left| H(\Omega_k, \theta) \right|^2.$$

These changes reduce the cost function, within a parameter independent constant λ to

$$V_F(\theta) = \frac{1}{F} \sum_{k=1}^{F} \frac{\left| \hat{Y}(k) - G(\Omega_k, \theta) U_0(k) - T_G(\Omega_k, \theta) \right|^2}{\left| H(\Omega_k, \theta) \right|^2},$$

which is exactly the same expression as Eq. 37 in ▶ System Identification: An Overview, provided that the frequencies Ω_k cover the full unit circle and the transient term T_G is omitted. The latter models the initial condition effects in the time domain. The expression shows the full equivalence with the classical discrete time domain formulation. If only a subsection of the full unit circle is used for the fit, the additional term $N \log \det \Lambda$ in ▶ System Identification: An Overview, Eq. 21 should be added to the cost function $V_F(\theta)$.

Additional User Aspects in Parametric Frequency Domain System Identification

In this section we highlight some additional user aspects that are affected by the choice for a time or frequency approach to system identification.

Nonparametric Noise Models

The use of a nonparametric noise model is a natural choice in the frequency domain. It is of course also possible to use the parametric noise model in the frequency domain formulation, but then we would lose two major advantages of the frequency domain formulation: (i) For periodic excitations, there is no interaction between the identification of the plant model and the nonparametric noise model. Plant model errors do not show up in the noise model. (ii) The availability of the nonparametric noise models eliminates the need for tuning the parametric noise model order, resulting in algorithms that are easier to use.

A disadvantage of using a nonparametric noise model is that we can no longer express that the noise model can share some dynamics with the plant model, for example, when the disturbance is an unobserved plant input.

It is also possible to use a nonparametric noise model in the time domain. This leads to a Toeplitz weighting matrix, and the fast numerical algorithms that are used to deal with these make use internally of FFT (fast Fourier transform) algorithms which brings us back to the frequency domain representation of the data.

Stable and Unstable Plant Models

In the frequency domain formulation, there is no special precaution needed to deal with unstable models, so that we can tolerate these models without any problem. There are multiple reasons why this can be advantageous. The most obvious one is the identification of an unstable system, operating in a stabilizing closed loop. It can also happen that the intermediate models that are obtained during the optimization process are unstable. Imposing stability at each iteration can be too restrictive, resulting in estimates that are trapped in a local minimum. A last significant advantage is the possibility to split the identification problem (extract a model from the noisy data) and the approximation problem (approximate the unstable model by a stable one). This allows us to use in each step a cost function that is optimal for that step: maximum noise reduction in the first step, followed by a user-defined approximation criterion in the second step.

Model Selection and Validation

An important step in the system identification procedure is the tuning of the model complexity, followed by the evaluation of the model quality on a fresh data set. The availability of a

nonparametric noise model and a high-quality frequency response function measurement simplifies these steps significantly:

Absolute Interpretation of the Cost Function: Impact on Model Selection and Validation

The weighted least squares cost function, using the nonparametric noise weighting, is a normalized function. Its expected value equals $E\{V(\hat{\theta})\} = (F - n_\theta/2)/F$, with n_θ the number of free parameters in the model and F the number of frequency points. A cost function that is far too large points to remaining model errors. Unmodeled dynamics result in correlated residuals (difference between the measured and the modeled FRF): the user should increase the model order to capture these dynamics in the linear model. A cost function that is far too large, while the residuals are white, points to the presence of nonlinear distortions: the best linear approximation is identified, but the user should be aware that this approximation is conditioned on the actual excitation signal. A cost function that is far too low points to an error in the preprocessing of the data, resulting in a bad noise model.

Missing Resonances

Some systems are lightly damped, resulting in a resonant behavior, for example, a vibrating mechanical structure. By comparing the parametric transfer function model with the nonparametric FRF measurements, it becomes clearly visible if a resonance is missed in the model. This can be either due to a too simple model structure (the model order should be increased), or it can appear because the model is trapped in a local minimum. In the latter case, better numerical optimization and initialization procedures should be looked for.

Identification in the Presence of Noise on the Input and Output Measurements

Within the band-limited measurement setup, both the input and the output have to be measured. This leads in general to an identification framework where both the input and the output are disturbed by noise. Such problems are studied in the errors-in-variables (EIV)

framework (Soderstrom 2012). A special case is the identification of a system that is captured in a feedback loop. In that case we have that the noisy output measurements are fed back to the input of the system which creates a dependency between the input and output disturbance. We discuss both situations briefly below.

Errors-in-Variables Framework

The major difficulty of the EIV framework is the simultaneous identification of the plant model describing the input-output relations, the noise models that describe the input and the output noise disturbances, and the signal model describing the coloring of the excitation (Soderstrom 2012). Advanced identification methods are developed, but today it is still necessary to impose strong restrictions on the noise models, e.g., correlations between input and output noise disturbances are not allowed. The periodic frequency domain approach encapsulates the general EIV, including mutually correlated colored input–output noise. Again, a full nonparametric noise model is obtained in the preprocessing step. This reduces the complexity of the EIV problem to that of a classical weighted least squares identification problem which makes a huge difference in practice (Pintelon and Schoukens 2012; Söderström et al. 2010).

Identification in a Feedback Loop

Identification under feedback conditions can be solved in the time domain prediction error method (Ljung 1999, Sect. 13.4; Söderström and Stoica 1989, Chap. 10). This leads to consistent estimates, provided that the exact plant and noise model structure and order is retrieved. In the periodic frequency domain approach, a nonparametric noise model is extracted (variances input and output noise, and covariance between input and output noise) in the preprocessing step, without any user interaction. Next, these are used as a weighting in the weighted least squares cost function which leads to consistent estimates provided that the plant model is flexible enough to capture the true plant

transfer function (Pintelon and Schoukens 2012, Sect. 9.18).

Summary and Future Directions

Theoretically, there is a full equivalence between the time and frequency domain formulation of the system identification problem. In many practical situations the user can make a free choice between both approaches, based on nontechnical arguments like familiarity with one of both domains. However, some problems can be easier formulated in the frequency domain. Identification of continuous time models is not more involved than retrieving a discrete time model. The frequency domain formulation is also the natural choice to use nonparametric noise models. This eliminates the request to select a specific noise model structure and order, although this might be a drawback for experienced users who can take advantage of a clever choice of this structure. The advantages that are directly linked to periodic excitation signals can be explored most naturally in the frequency domain: the noise model is available for free, EIV identification is not more involved than the output error identification problem, and identification under feedback conditions does not differ from open-loop identification. Nonstationary effects are an example of a problem that will be easier detected in the time domain. In general, we advise the reader to take the best of both approaches and to swap from one domain to the other whenever it gives some advantage to do so. In the future, it will be necessary to extend the framework to include a characterization of nonlinear and time-varying effects.

Cross-References

▶ System Identification: An Overview
▶ Nonparametric Techniques in System Identification
▶ Experiment Design and Identification for Control

Recommended Reading

We recommend the reader the books of Ljung (1999) and Söderström and Stoica (1989) for a systematic study of time domain system identification. The book of Pintelon and Schoukens (2012) gives a comprehensive introduction to frequency domain identification. An extended discussion of the basic choices (intersample behavior, measurement setup) is given in Chap. 13 of Pintelon and Schoukens (2012) or in Schoukens et al. (1994). The other references in this list highlight some of the technical aspects that were discussed in this text.

Bibliography

Agüero JC, Yuz JI, Goodwin GC et al (2009) On the equivalence of time and frequency domain maximum likelihood estimation. Automatica 46:260–270

Goodwin GC (2010) Sampling and sampled-data models. In: American control conference, ACC 2010, Baltimore

Kollar I (1994) Frequency domain system identification toolbox for use with MATLAB. The MathWorks Inc., Natick

Ljung L (1988) System identification toolbox for use with MATLAB. The MathWorks Inc., Natick

Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice Hall, Upper Saddle River

Ljung L (2006) Frequency domain versus time domain methods in system identification – revisited. In: Workshop on control of uncertain systems location, University of Cambridge, 21–22 Apr

Ljung L, Glover K (1981) Frequency-domain versus time domain methods in system identification. Automatica 17:71–86

McKelvey T (2002) Frequency domain identification methods. Circuits Syst Signal Process 21: 39–55

Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. Wiley, Hoboken/IEEE, Piscataway

Pintelon R, Schoukens J, Vandersteen G (1997) Frequency domain system identification using arbitrary signals. IEEE Trans Autom Control 42:1717–1720

Pintelon R, Schoukens J, Pauwels L et al (2005) Diffusion systems: stability, modeling, and identification. IEEE Trans Instrum Meas 54:2061–2067

Schoukens J, Pintelon R, Van hamme H (1994) Identification of linear dynamic systems using piecewise-constant excitations – use, misuse and alternatives. Automatica 30:1153–1169

Schoukens J, Pintelon R, Vandersteen G et al (1997) Frequency-domain system identification using nonparametric noise models estimated from a small number of data sets. Automatica 33: 1073–1086

Schoukens J, Pintelon R, Rolain Y (1999) Study of conditional ML estimators in time and frequency-domain system identification. Automatica 35:91–100

Söderström T (2012) System identification for the errors-in-variables problem. Trans Inst Meas Control 34:780–792

Söderström T, Stoica P (1989) System identification. Prentice-Hall, Englewood Cliffs

Söderström T, Hong M, Schoukens J et al (2010) Accuracy analysis of time domain maximum likelihood method and sample maximum likelihood method for errors-in-variables and output error identification. Automatica 46:721–727

Frequency-Response and Frequency-Domain Models

Abbas Emami-Naeini and J. David Powell
Stanford University, Stanford, CA, USA

Abstract

A major advantage of using frequency response is the ease with which experimental information can be used for design purposes. Raw measurements of the output amplitude and phase of a plant undergoing a sinusoidal input excitation are sufficient to design a suitable feedback control. No intermediate processing of the data (such as finding poles and zeros or determining system matrices) is required to arrive at the system model. The wide availability of computers has rendered this advantage less important now than it was years ago; however, for relatively simple systems, frequency response is often still the most cost-effective design method. The method is most effective for systems that are stable in open-loop. Yet another advantage is that it is the easiest method to use for designing dynamic compensation.

Keywords

Introduction: Frequency Response

A very common way to use the exponential response of linear time-invariant systems (LTIs) is in finding the **frequency response,** or response to a sinusoid. First we express the sinusoid as a sum of two exponential expressions (Euler's relation):

$$A\cos(\omega t) = \frac{A}{2}(e^{j\omega t} + e^{-j\omega t}). \qquad (1)$$

Suppose we have an LTI system with input u and output y. If we let $s = j\omega$ in the transfer function $G(s)$, then the response to $u(t) = e^{j\omega t}$ is $y(t) = G(j\omega)e^{j\omega t}$; similarly, the response to $u(t) = e^{-j\omega t}$ is $G(-j\omega)e^{-j\omega t}$. By superposition, the response to the sum of these two exponentials, which make up the cosine signal, is the sum of the responses:

$$y(t) = \frac{A}{2}[G(j\omega)e^{j\omega t} + G(-j\omega)e^{-j\omega t}]. \quad (2)$$

The transfer function $G(j\omega)$ is a complex number that can be represented in polar form or in magnitude-and-phase form as $G(j\omega) = M(\omega)e^{j\phi(\omega)}$, or simply $G = Me^{j\phi}$. With this substitution, Eq. (2) becomes for a specific input frequency $\omega = \omega_o$

$$y(t) = \frac{A}{2}M\left(e^{j(\omega t + \varphi)} + e^{-j(\omega t + \varphi)}\right),$$

$$= AM\cos(\omega t + \varphi), \qquad (3)$$

$$M = |G(j\omega)| = |G(s)|_{s=j\omega_o}$$

$$= \sqrt{\{\text{Re}[G(j\omega_o)]\}^2 + \{\text{Im}[G(j\omega_o)]\}^2},$$

$$\varphi = \angle G(j\omega) = \tan^{-1}\left[\frac{\text{Im}[G(j\omega_o)]}{\text{Re}[G(j\omega_o)]}\right]. \quad (4)$$

This means that if an LTI system represented by the transfer function $G(s)$ has a sinusoidal input with magnitude A, the output will be sinusoidal at the *same* frequency with magnitude AM and will be shifted in phase by the angle φ. M is usually referred to as the **amplitude ratio** or **magnitude** and φ is referred to as the **phase** and they are both functions of the input frequency, ω. The frequency response can be measured experimentally quite easily in the laboratory by driving the system with a known sinusoidal input, letting the transient response die, and measuring the steady-state amplitude and phase of the system's output as shown in Fig. 1. The input frequency is set to sufficiently many values so that curves such as the one in Fig. 2 are obtained. Bode suggested that we plot $\log|M|$ vs. $\log\omega$ and $\varphi(\omega)$ vs. $\log\omega$ to best show the essential features of $G(j\omega)$. Hence, such plots are referred to as Bode plots. Bode plotting techniques are discussed in Franklin et al. (2015).

We are interested in analyzing the frequency response not only because it will help us understand how a system responds to a sinusoidal input, but also because evaluating $G(s)$ with s taking on values along the $j\omega$ axis will prove to be very useful in determining the stability of a closed-loop system. Since the $j\omega$ axis is the boundary between stability and instability; evaluating $G(j\omega)$ provides information that allows us to determine closed-loop stability from the open-loop $G(s)$.

For the second-order system

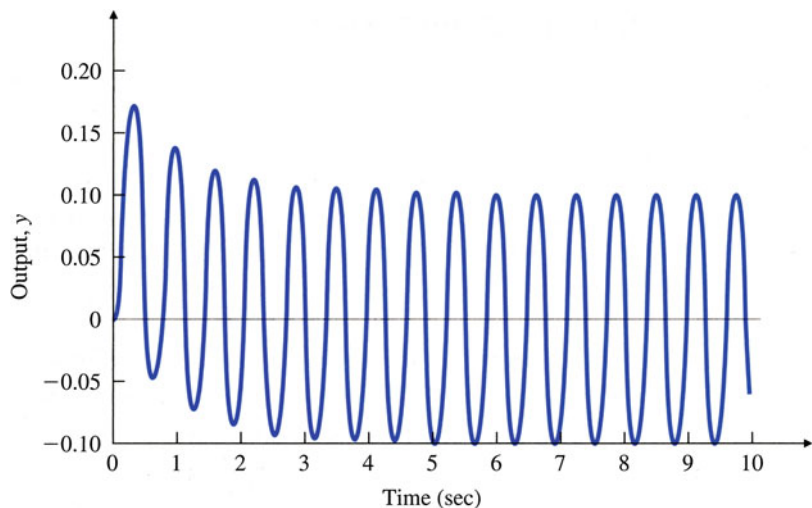$$G(s) = \frac{1}{(s/\omega_n)^2 + 2\zeta(s/\omega_n) + 1}, \quad (5)$$

the Bode plot is shown in Fig. 3 for various values of ζ.

A natural specification for system performance in terms of frequency response is the **bandwidth**, defined to be the maximum frequency at which the output of a system will track an input sinusoid in a satisfactory manner. By convention, for the system shown in Fig. 4 with a sinusoidal input r, the bandwidth is the frequency of r at which the output y is attenuated to a factor of 0.707 times the input (If the output is a voltage across a 1-Ω resistor, the power is v^2 and when $|v| = 0.707$, the power is reduced by a factor of 2. By convention, this is called the half-power point.). Figure 5 depicts the idea graphically for the frequency response of the *closed-loop* transfer function
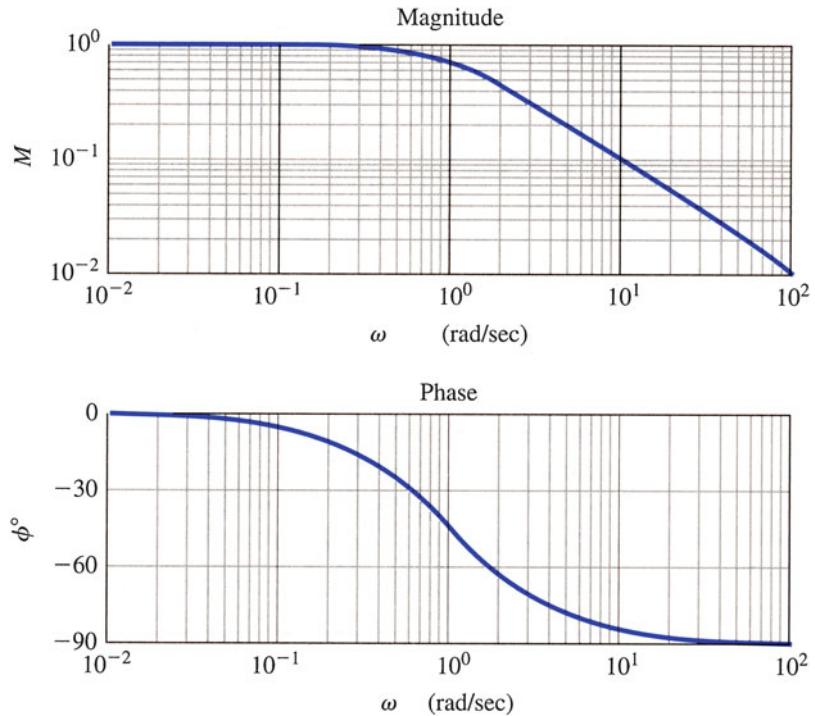
$$\frac{Y(s)}{R(s)} \triangleq \mathcal{T}(s) = \frac{KG(s)}{1 + KG(s)}. \quad (6)$$

Frequency-Response and Frequency-Domain Models, Fig. 1 Response of $G(s) = \frac{1}{(s+1)}$ to the input $u = \sin 10t$ (Source: Franklin et al. (2010), p. 298, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

The plot is typical of most closed-loop systems in that (1) the output follows the input ($|\mathcal{T}| \cong 1$) at the lower excitation frequencies and (2) the output ceases to follow the input ($|\mathcal{T}| < 1$) at the higher excitation frequencies. The maximum value of the frequency-response magnitude is referred to as the **resonant peak M_r**.

Bandwidth is a measure of speed of response and is therefore similar to time-domain measures such as rise time and peak time or the s-plane measure of dominant-root(s) natural frequency. In fact, if the $KG(s)$ in Fig. 4 is such that the closed-loop response is given by Fig. 3a, we can see that the bandwidth will equal the natural frequency of the closed-loop root (that is, $\omega_{BW} = \omega_n$ for a closed-loop damping ratio of $\zeta = 0.7$). For other damping ratios, the bandwidth is approximately equal to the natural frequency of the closed-loop roots, with an error typically less than a factor of 2.

For a second-order system, the time responses are functions of the pole-location parameters ζ and ω_n. If we consider the curve for $\zeta = 0.5$ to

be an average, the rise time (Rise time t_r.) from $y = 0.1$ to $y = 0.9$ is approximately $\omega_n t_r = 1.8$. Thus, we can say that

$$t_r \cong \frac{1.8}{\omega_n}. \tag{7}$$

Although this relationship could be embellished by including the effect of the damping ratio, it is important to keep in mind how Eq. (7) is typically used. It is accurate only for a second-order system with no zeros; for all other systems it is a rough approximation to the relationship between t_r and ω_n. Most systems being analyzed for control systems design are more complicated than the pure second-order system, so designers use Eq. (7) with the knowledge that it is a rough approximation only. Hence, for a second-order system the bandwidth is inversely proportional to the rise time, t_r. Hence we are able to link the time and frequency domain quantities in this way.

The definition of the bandwidth stated here is meaningful for systems that have a low-pass filter behavior, as is the case for any physical control

Frequency-Response and Frequency-Domain Models, Fig. 3 Frequency responses of standard second-order systems (**a**) magnitude (**b**) phase (Source: Franklin et al. (2010), p. 303, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)
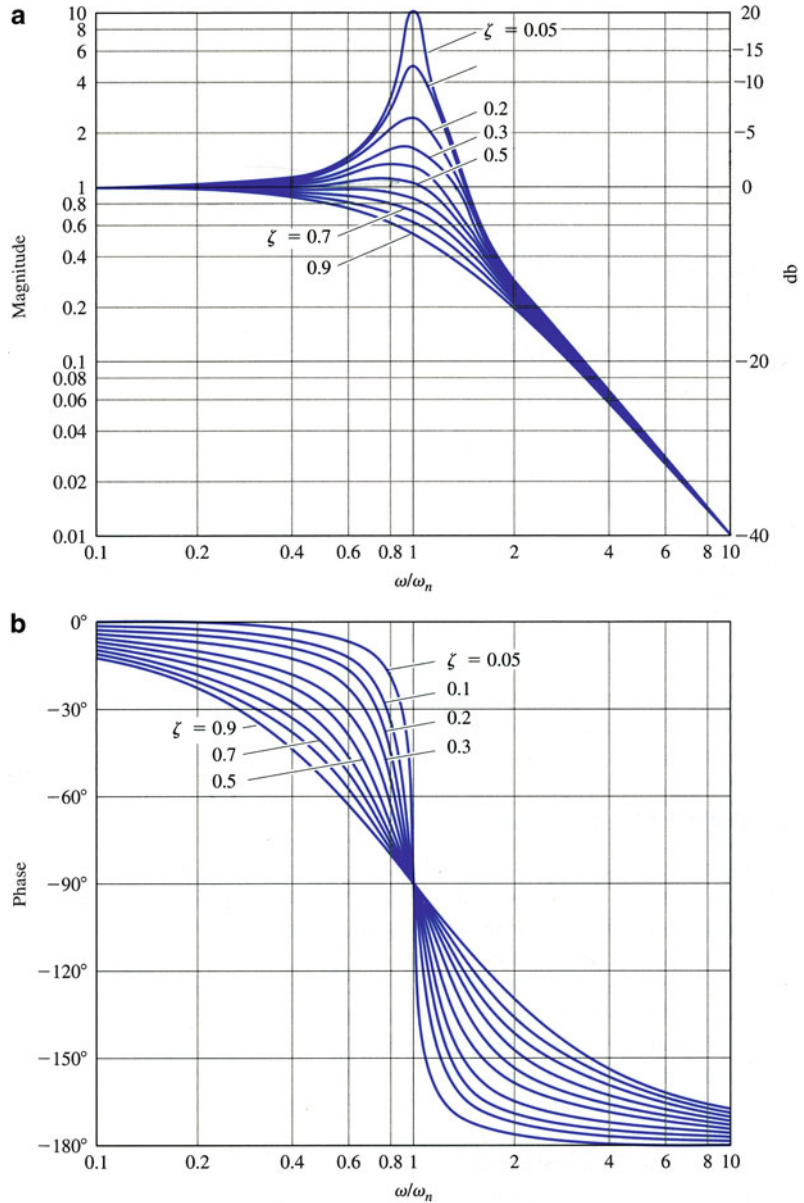
system. In other applications the bandwidth may be defined differently. Also, if the ideal model of the system does not have a high-frequency roll-off (e.g., if it has an equal number of poles and zeros), the bandwidth is infinite; however, this does not occur in nature as nothing responds well at infinite frequencies.
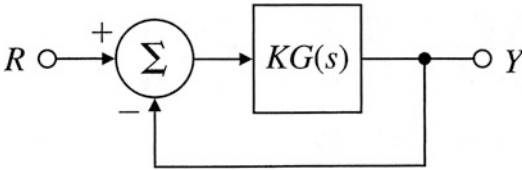
In many cases, the designer's primary concern is the error in the system due to disturbances rather than the ability to track an input. For error

analysis, we are more interested in the sensitivity function $\mathcal{S}(s) = 1 - \mathcal{T}(s)$, rather than $\mathcal{T}(s)$. For most open-loop systems with high gain at low frequencies, $\mathcal{S}(s)$ for a disturbance input has very low values at low frequencies and grows as the frequency of the input or disturbance approaches the bandwidth. For analysis of either $\mathcal{T}(s)$ or $\mathcal{S}(s)$, it is typical to plot their response versus the frequency of the input. Either frequency response for control systems design can be evaluated using

the computer or can be quickly sketched for simple systems using the efficient methods described in Franklin et al. (2015). The methods described next are also useful to expedite the

design process as well as to perform sanity checks on the computer output.

Neutral Stability: Gain and Phase Margins

In the early days of electronic communications, most instruments were judged in terms of their frequency response. It is therefore natural that when the feedback amplifier was introduced, techniques to determine stability in the presence of feedback were based on this response.

Suppose the closed-loop transfer function of a system is known. We can determine the stability of a system by simply inspecting the
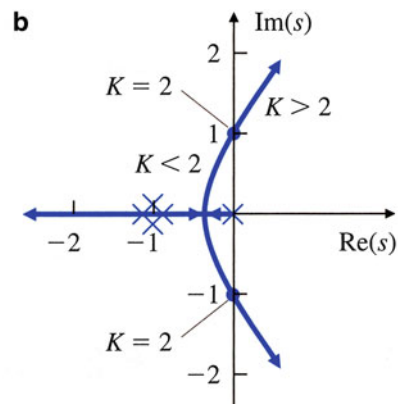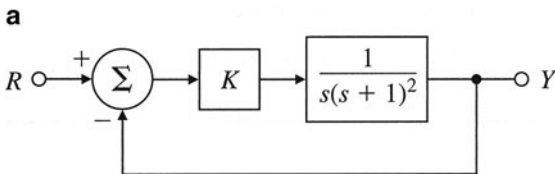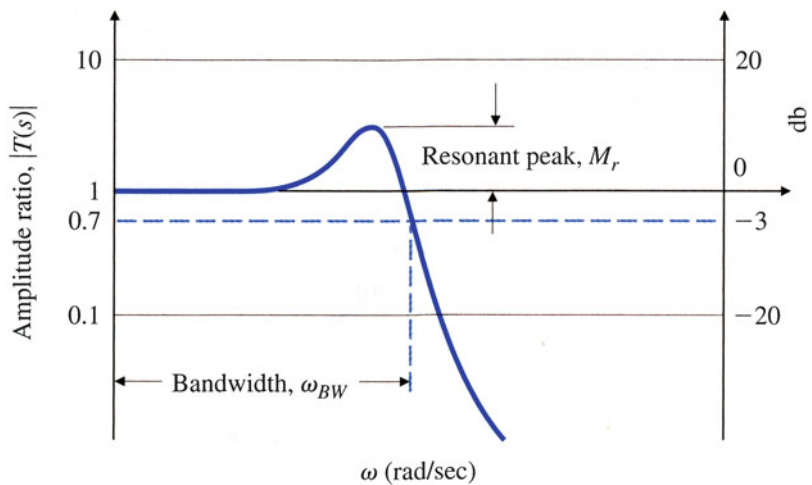

Frequency-Response and Frequency-Domain Models, Fig. 4 Unity feedback system (Source: Franklin et al. (2010), p. 304, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

Frequency-Response and Frequency-Domain Models, Fig. 5 Definitions of bandwidth and resonant peak (Source: Franklin et al. (2010), p. 304, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

Frequency-Response and Frequency-Domain Models, Fig. 6 Stability example: (**a**) system definition; (**b**) root locus (Source: Franklin et al. (2010), p. 318, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

denominator in factored form (because the factors give the system roots directly) to observe whether the real parts are positive or negative. However, the closed-loop transfer function is usually not known. In fact, the whole purpose behind understanding the root-locus technique is to be able to find the factors of the denominator in the closed-loop transfer function, given only the open-loop transfer function. Another way to determine closed-loop stability is to evaluate the frequency response of the *open-loop* transfer function $KG(j\omega)$ and then perform a test on that response. Note that this method also does not require factoring the denominator of the closed-loop transfer function. In this section we will explain the principles of this method. Note that this method also does not require factoring the denominator of the closed-loop transfer function. Here we will explain the principles of this method.

Suppose we have a system defined by Fig. 6a and whose root locus behaves as shown in Fig. 6b; that is, instability results if K is larger than 2. The neutrally stable points lie on the imaginary axis – that is, where $K = 2$ and $s = j\,1.0$. Furthermore, all points on the root locus have the property that

$$|KG(s)| = 1 \quad \text{and} \quad \angle G(s) = -180°.$$

At the point of neutral stability we see that these root-locus conditions hold for $s = j\omega$, so

$$|KG(j\omega)| = 1 \quad \text{and} \quad \angle G(j\omega) = -180°. \tag{8}$$

Thus, a Bode plot of a system that is neutrally stable (that is, with K defined such that a closed-loop root falls on the imaginary axis) will satisfy the conditions of Eq. (8). Figure 7 shows the frequency response for the system whose root locus is plotted in Fig. 6 for various values of K. The magnitude response corresponding to $K = 2$ passes through 1 at the same frequency ($\omega = 1$ rad/s) at which the phase passes through $-180°$, as predicted by Eq. (8).

Having determined the point of neutral stability, we turn to a key question: Does increasing the gain increase or decrease the system's stability?

We can see from the root locus in Fig. 6b that any value of K less than the value at the neutrally stable point will result in a stable system. At the frequency ω where the phase $\angle G(j\omega) = -180°$ ($\omega = 1$ rad/s), the magnitude $|KG(j\omega)| < 1.0$ for stable values of K and >1 for unstable values of K. Therefore, we have the following trial stability condition, based on the character of the open-loop frequency response:

$$|KG(j\omega)| < 1 \quad \text{at} \quad \angle G(j\omega) = -180°. \tag{9}$$

This stability criterion holds for all systems for which increasing gain leads to instability and $|KG(j\omega)|$ crosses the magnitude ($=1$) once, the most common situation. However, there are systems for which an increasing gain can lead from instability to stability; in this case, the stability condition is

$$|KG(j\omega)| > 1 \quad \text{at} \quad \angle G(j\omega) = -180°. \tag{10}$$

Based on the above ideas, we can now define the robustness metrics gain and phase margins:

Phase Margin: Suppose at ω_1, $|G(j\omega_1)| = \frac{1}{K}$. How much more *phase* could the system tolerate (as a time delay, perhaps) before reaching the stability boundary? The answer to this question follows from Eq. (8), i.e., the phase margin (PM) is defined as

$$\text{PM} = \angle G(j\omega_1) - (-180°). \tag{11}$$

**Gain Margin:** Suppose at ω_2, $\angle G(j\omega_2) = -180°$. How much more *gain* could the system tolerate (as an amplifier, perhaps) before reaching the stability boundary? The answer to this question follows from Eq. (9), i.e., the gain margin (GM) is defined as

$$\text{GM} = \frac{1}{K|G(j\omega_2)|}. \tag{12}$$

There are also rare cases when $|KG(j\omega)|$ crosses magnitude ($=1$) more than once, or where an increasing gain leads to instability. A rigorous way to resolve these situations is to use the Nyquist

Frequency-Response and Frequency-Domain Models, Fig. 7 Stability example: (**a**) system definition; (**b**) root locus (Source: Franklin et al. (2010), p. 319, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)
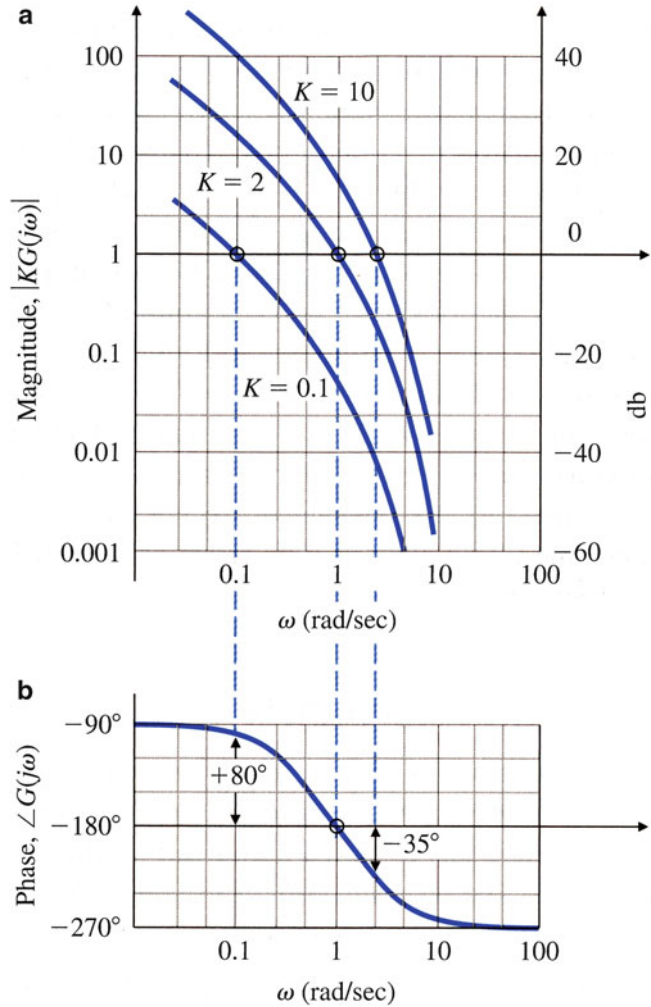
stability criterion as discussed in Franklin et al. (2015).
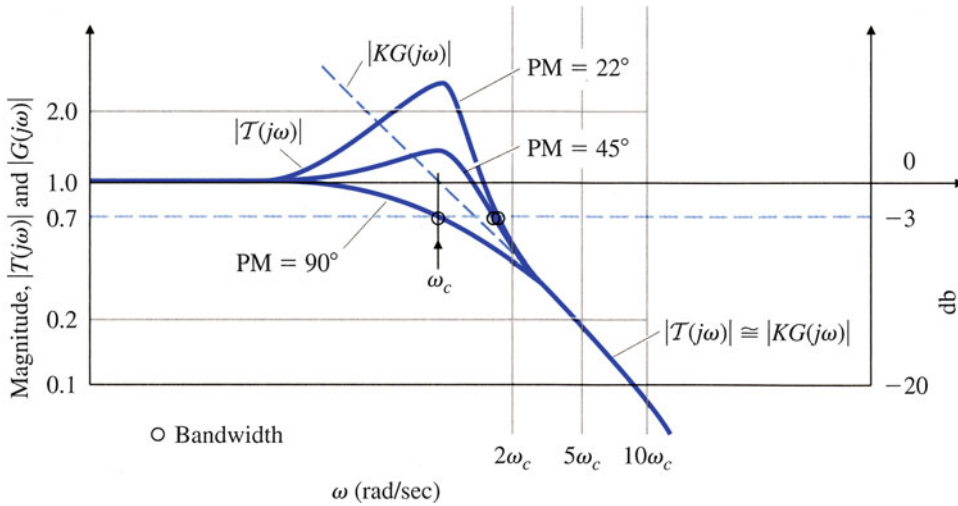
Closed-Loop Frequency Response

The closed-loop bandwidth was defined earlier in this section. The natural frequency is always within a factor of 2 of the bandwidth for a second-order system. We can help establish a more exact correspondence by making a few observations. Consider a system in which $|KG(j\omega)|$ shows the typical behavior

$$|KG(j\omega)| \gg 1 \quad \text{for} \quad \omega \ll \omega_c,$$

$$|KG(j\omega)| \ll 1 \quad \text{for} \quad \omega \gg \omega_c,$$

where ω_c is the crossover frequency. The closed-loop frequency-response magnitude is approximated by

$$|\mathcal{T}(j\omega)| = \left| \frac{KG(j\omega)}{1 + KG(j\omega)} \right| \cong \begin{cases} 1, & \omega \ll \omega_c, \\ |KG|, & \omega \gg \omega_c. \end{cases} \tag{13}$$

In the vicinity of crossover, where $|KG(j\omega)| = 1$, $|\mathcal{T}(j\omega)|$ depends heavily on the PM. A PM of 90° means that $\angle G(j\omega_c) = -90°$, and therefore $|\mathcal{T}(j\omega_c)| = 0.707$. On the other hand, PM = 45° yields $|\mathcal{T}(j\omega_c)| = 1.31$.

The exact evaluation of Eq. (13) was used to generate the curves of $|\mathcal{T}(j\omega)|$ in Fig. 8. It shows that the bandwidth for smaller values of PM is typically somewhat greater than ω_c, though usually it is less than $2\omega_c$; thus,

Frequency-Response and Frequency-Domain Models, Fig. 8 Closed-loop bandwidth with respect to PM (Source: Franklin et al. (2010), p. 347, reprinted by permission of Pearson Education, Inc., Upper Saddle River, NJ)

$$\omega_c \leq \omega_{BW} \leq 2\omega_c. \qquad (14)$$

Another specification related to the closed-loop frequency response is the resonant-peak magnitude M_r, defined in Fig. 5. For linear systems, M_r is generally related to the damping of the system. In practice, M_r is rarely used; most designers prefer to use the PM to specify the damping of a system, because the imperfections that make systems nonlinear or cause delays usually erode the phase more significantly than the magnitude.

It is also important in the design to achieve certain error characteristics and these are often evaluated as a function of the input or disturbance frequency. In some cases, the primary function of the control system is to regulate the output to a certain constant input in the presence of disturbances. For these situations, the key item of interest for the design would be the closed-loop frequency response of the error with respect to disturbance inputs.

Summary and Future Directions

The frequency response methods are the most popular because they can deal with model uncertainty and can be measured in the laboratory. A wide range of information about the system can be displayed in a Bode plot. The dynamic compensation can be carried out directly from the Bode plot. Extension of the ideas to multivariable systems has been done via singular value plots. Extension to nonlinear systems is still the subject of current research.

Cross-References

▶ Classical Frequency-Domain Design Methods
▶ Frequency Domain System Identification

Bibliography

Franklin GF, Powell JD, Emami-Naeini A (2015) Feedback control of dynamic systems, 7th edn. Pearson Education, Upper Saddle River, NJ, Boston
Franklin GF, Powell JD, Emami-Naeini A (2010) Feedback control of dynamic systems, 6th edn. Pearson Education, Upper Saddle River

FTC

▶ Fault-Tolerant Control

Fundamental Limitation of Feedback Control

Jie Chen
City University of Hong Kong, Hong Kong,
China

Abstract

Feedback systems are designed to meet many different objectives. Yet, not all design objectives are achievable as desired due to the fact that they are often mutually conflicting and that the system properties themselves may impose design constraints and thus limitations on the performance attainable. An important step in the control design process is then to analyze what and how system characteristics may impose constraints, and accordingly, how to make tradeoffs between different objectives by judiciously navigating between the constraints. Fundamental limitation of feedback control is an area of research that addresses these constraints, limitations, and tradeoffs.

Keywords

Bode integrals; Design tradeoff; Performance limitation; Tracking and regulation limits

Introduction

Fundamental control limitations are referred to those intrinsic of feedback that can neither be overcome nor circumvent regardless how it may be designed. By this nature, the study of fundamental limitations dwells on a Hamletian question: Can or can't it be done? What can and cannot be done? To be more specific, yet still general enough, at heart here are issues concerning the benefit and cost of feedback. We ask such questions as (1) What system characteristics may impose inherent limitations regardless of controller design? (2) What inherent constraints may exist in design, what kind of tradeoffs are to be made? (3) What are the best achievable performance limits? (4) How can the constraints, limitations, and limits be quantified, in ways meaningful for control analysis and design? Needless to say, issues of this kind are very general and in fact are commonplace in science and engineering. Analogies can be made, for example, to Shannon's theorems in communications theory, the Cramer-Rao bound in statistics, and Heisenberg's uncertainty principle in quantum mechanics; they all address the fundamental limits and limitations, though for different problems and in different contexts. The search for fundamental limitations of feedback control, as such, may be considered a quest for an "ultimate truth" or the "law of feedback."

For their fundamentality and importance, inquiries into control performance limitations have persisted over time and continue to be of vital interest. It is worth emphasizing, however, that performance limitation studies are not merely driven by intellectual curiosity, but are tantamount to better and more realistic feedback systems design and hence of tangible practical value. An analysis of performance limitations can aid control design in several aspects. First, it may provide a fundamental limit on the best performance attainable irrespective of controller design, thus furnishing a guiding benchmark in the design process. Secondly, it helps a designer assess what and how system properties may be inherently conflicting and thus pose inherent difficulties to performance objectives, which in turn helps the designer specify reasonable goals, and make judicious modifications and revisions on the design. In this process, the theory of fundamental control limitations promises to provide valuable insights and analytical justifications to long-held design heuristics and, indeed, to extend such heuristics further beyond. This has become increasingly more relevant, as modern control design theory and practice relies heavily on optimization-based numerical routines and tools.

Systematic investigation and understanding of fundamental control limitations began with the classical work of Bode in the 1940s on

logarithmic sensitivity integrals, known as the Bode integrals. Bode's work has had a lasting impact on the theory and practice of control and has inspired continued research effort dated most recently, leading to a variety of extensions and new results which seek to quantify design constraints and performance limitations by logarithmic integrals of Bode and Poisson type. On the other hand, the search for the best achievable performance is a natural goal in optimal control problems, which has lend bounds on optimal performance indices defined under various criteria. Likewise, the latter developments have also been substantial and are continuing to branch to different problems and different system categories.

In this entry we attempt to provide a summary overview of the key developments in the study of fundamental limitations of feedback control. While the understanding on this subject has been compelling and the results are rather prolific, we focus on Bode-type integral relations and the best achievable performance limits, two branches of the study that are believed to be most well-developed. Roughly speaking, the Bode-type integrals are most useful for quantifying the inherent design constraints and tradeoffs in the frequency domain, while the performance results provide fundamental limits of canonical control objectives defined using frequency- and time-domain criteria. Invariably, the two sets of results are intimately related and reinforce each other. The essential message then is that despite its many benefits, feedback has its own limitations and is subject to various constraints. Feedback design, for that sake, requires often times a hard tradeoff.

Control Design Specifications

We begin by introducing the basic notation to be used in the sequel. Let $\mathbb{C}_+ := \{z : \text{Re}(z) > 0\}$ denote the open right half plane (RHP) and $\overline{\mathbb{C}}_+$ the closed RHP (CRHP). For a complex number z, we denote its conjugate by $\overline{z}$. For a complex vector x, we denote its conjugate transpose by x^H, and its Euclidean norm by $\|x\|_2$. The largest

singular value of a matrix A will be written as $\overline{\sigma}(A)$. If A is a Hermitian matrix, we denote by $\overline{\lambda}(A)$ its largest eigenvalue. For any unitary vectors $u, v \in \mathbb{C}^n$, we denote by $\angle(u, v)$ the *principal angle* between the two one-dimensional subspaces, called the directions, spanned by u and v:

$$\cos \angle(u, v) := |u^H v|.$$

For a stable continuous-time system with transfer function matrix $G(s)$, we define its $\mathcal{H}_\infty$ norm by

$$\|G\|_\infty := \sup_{\text{Re}(s) > 0} \overline{\sigma}(G(s)).$$

We consider the standard configuration of finite-dimensional linear time-invariant (LTI) feedback control systems given in Fig. 1. In this setup, P and K represent the transfer functions of the plant model and controller, respectively, r is a command signal, d a disturbance, n a noise signal, and y the output response. Define the open-loop transfer function, the sensitivity function, and the complementary sensitivity function by

$$L = PK, \quad S = (I + L)^{-1}, \quad T = L(I + L)^{-1},$$

respectively. Then the output can be expressed as

$$y = Sd - Tn + SPr.$$

The goal of feedback control design is to design a controller K so that the closed-loop system is stable and that it achieves certain performance specifications. Typical design objectives include:

- *Disturbance attenuation*. The effect of the disturbance signal on the output should be kept small, which translates into the require-
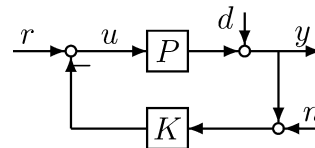
Fundamental Limitation of Feedback Control, Fig. 1
Feedback configuration

ment that the sensitivity function be small in magnitude at the frequencies of interest. For a single-input single-output (SISO) system, this mandates that

$$|S(j\omega)| < 1, \quad \forall \, \omega \in [0, \, \omega_1).$$

The sensitivity magnitude $|S(j\omega)|$ is to be kept as small as possible in the low frequency range.

- *Noise reduction.* The noise response should be reduced at the output. This requires that the complementary sensitivity function be small in magnitude at frequencies of interest. For a SISO system, the objective is to achieve

$$|T(j\omega)| < 1, \quad \forall \, \omega \in [\omega_2, \, \infty).$$

Similarly, the magnitude $|T(j\omega)|$ is desired to be the smallest at high frequencies.

Moreover, feedback can be introduced to achieve many other objectives including regulation, command tracking, improved sensitivity to parameter variations, and, more generally, system robustness, all by manipulating the three key transfer functions: the open-loop transfer function, the sensitivity function, and the complementary sensitivity function.

The design and implementation of feedback systems, on the other hand, are also subject to many constraints, which include

1. Causality: A system must be causal for it to be implementable. This constraint requires that no ideal filter can be used for compensation and that the system's relative degree and delay be preserved.
2. Stability: The closed-loop system must be stable. This implies that every closed-loop transfer function must be bounded and analytic in CRHP.
3. Interpolation: There should be no unstable pole-zero cancelation between the plant and controller, in order to rid of hidden instability. Thus, at each RHP pole p_i and zero z_i, it is necessary that

$$S(p_i) = 0, \qquad T(p_i) = 1,$$

$$S(z_i) = 1, \qquad T(z_i) = 0.$$

4. Structural constraints: Constraints in this category arise from the feedback structure itself; for example, $S(s) + T(s) = 1$. The implication then is that the closed-loop transfer functions cannot be independently designed, thus resulting in conflicting design objectives.

For a given plant, each of these constraints is unalterable and hence is fundamental, and each will constrain the performance attainable in one way or another. The question we face then is how the constraints may be captured in a form that is directly pertinent and useful to feedback design.

Bode Integral Relations

In the classical feedback control theory, Bode's gain-phase formula (Bode 1945) is used to express the aforementioned design constraints for SISO systems.

Bode Gain-Phase Integral *Suppose that $L(s)$ has no pole and zero in $\overline{\mathbb{C}}_+$. Then at any frequency ω_0,*

$$\angle L(j\omega_0) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d \log |L|}{d\nu} \log \coth \frac{|\nu|}{2} d\nu.$$

A special form of the Hilbert transform, this gain-phase formula relates the gain and phase of the open-loop transfer function evaluated along the imaginary axis, whose implication may be explained as follows. In order to make the sensitivity response small in the low frequency range, the open-loop transfer function is required to have a high gain; the higher, the better. On the other hand, for noise reduction and robustness purposes, we need to keep the loop gain low at high frequencies, the lower the better. Evidently, to maximize these objectives, we want the two frequency bands as wide as possible. This then requires a steep decrease of the loop gain and hence a rather negative slope in the crossover region, say, the intermediate frequency range near ω_0. But the gain-phase relationship tells that a very negative derivative in the gain will lead to

a very negative phase, driving the phase closer to the negative 180 degree, namely, the critical point of stability. It consequently reduces the phase margin and may even cause instability. As a result, the gain-phase relationship demonstrates a conflict between the two design objectives. It is safe to claim that much of the classical feedback design theory came as a consequence of this simple relationship, aiming to shape the open-loop frequency response in the crossover region by trial and error, using lead or lag filters.

While in using the gain-phase formula, the design specifications imposed on closed-loop transfer functions are translated approximately into the requirements on the open-loop transfer function, and the tradeoff between different design goals is achieved by shaping the open-loop gain and phase; a more direct vehicle to accomplish this same goal is Bode's sensitivity integral (Bode 1945).

Bode Sensitivity Integrals *Let $p_i \in \mathbb{C}_+$ be the unstable poles and $z_i \in \mathbb{C}_+$ the nonminimum phase zeros of $L(s)$. Suppose that the closed-loop system in Fig. 1 is stable.*

(i) If $L(s)$ has relative degree greater than one, then

$$\int_0^\infty \log |S(j\omega)| d\omega = \pi \sum_i p_i.$$

(ii) If $L(s)$ contains no less than two integrators, then

$$\int_0^\infty \frac{\log |T(j\omega)|}{\omega^2} d\omega = \pi \sum_i \frac{1}{z_i}.$$

Bode's original work concerns the sensitivity integral for open-loop stable systems only. The integral relations shown herein, which are attributed to Freudenberg and Looze (1985) and Middleton (1991), respectively, provide generalizations to open-loop unstable and nonminimum phase systems.

Why are Bode sensitivity integrals important? What is the hidden message behind the mathematical formulas? Simply put, Bode integral exhibits that a feedback system's sensitivity must

abide some kind of conservation law, or invariance property: the integral of the logarithmic sensitivity magnitude over the entire frequency range must be a nonnegative constant, determined by the open-loop unstable poles. This property mandates a tradeoff between sensitivity reduction and sensitivity amplification in different frequency bands. Indeed, to achieve disturbance attenuation, the logarithmic sensitivity magnitude must stay below zero db, the lower the better. For noise reduction and robustness, however, its tail has to roll off sufficiently fast to zero db at high frequencies. Since, in light of the integral relation, the total area under the logarithmic magnitude curve is nonnegative, the logarithmic magnitude must rise above zero db, so that under its curve, the positive and negative areas may cancel each other to yield a nonnegative value. As such, an undesirable sensitivity amplification occurs, resulting in a fundamental tradeoff between the desirable sensitivity reduction and the undesirable sensitivity amplification, known colloquially as the *waterbed effect*.

MIMO Integral Relations

For a multi-input multi-output (MIMO) system depicted in Fig. 1, the sensitivity and complementary sensitivity functions, which now are transfer function matrices, satisfy similar interpolation constraints: at each RHP pole p_i and zero z_i of $L(s)$, the equations

$$S(p_i)\eta_i = 0, \qquad T(p_i)\eta_i = \eta_i,$$

$$w_i^H S(z_i) = w_i^H, \qquad w_i^H T(z_i) = 0$$

hold with some unitary vectors η_i and w_i, where η_i is referred to as a *right pole direction vector* associated with p_i, and w_i a *left zero direction vector* associated with z_i.

While it seems both natural and tempting, the extension of Bode integrals to MIMO systems has been highly nontrivial a task. Deep at the root is the complication resulted from the directionality properties of MIMO systems. Unlike in a SISO

system, the measure of frequency response magnitude is now the largest singular value of a transfer function matrix, which represents the worst-case amplification of energy-bounded signals, a direct counterpart to the gain of a scalar transfer function. This fact alone proves to cast a fundamental difference and poses a formidable obstacle. From a technical standpoint, the logarithmic function of the largest singular value is no longer a harmonic function as in the SISO case, but only a subharmonic function. Much to our regret then, familiar tools found from analytic function theory, such as Cauchy and Poisson theorems, the very backbone in developing Bode integrals, cease to be applicable. Nevertheless, it remains possible to extend Bode integrals in their essential spirit. Advances are made by Chen (1995, 1998, 2000).

MIMO Bode Sensitivity Integrals *Let $p_i \in \mathbb{C}_+$ be the unstable poles of $L(s)$ and $z_i \in \mathbb{C}_+$ the nonminimum phase zeros of $L(s)$. Suppose that the closed-loop system in Fig. 1 is stable.*

(i) If $L(s)$ has relative degree greater than one, then

$$\int_0^\infty \log \overline{\sigma}(S(j\omega))d\omega \geq \pi\overline{\lambda}\left(\sum_i p_i \eta_i \eta_i^H\right).$$

(ii) If $L(s)$ contains no less than two integrators, then

$$\int_0^\infty \frac{\log \overline{\sigma}(T(j\omega))}{\omega^2}d\omega \geq \pi\overline{\lambda}\left(\sum_i \frac{1}{z_i}w_i w_i^H\right)$$

where η_i and w_i are some unitary vectors related to the right pole direction vectors associated with p_i and the left zero direction vectors associated with z_i, respectively.

From these extensions, it is evident that same limitations and tradeoffs on the sensitivity and complementary sensitivity functions carry over to MIMO systems; in fact, both integrals reduce to the Bode integrals when specialized to

SISO systems. Yet there is something additional and unique of MIMO systems: the integrals now depend on not only the locations but also the directions of the zeros and poles. In particular, it can be shown that they depend on the mutual orientation of these directions, and the dependence can be explicitly characterized geometrically by the principal angles between the directions. This new phenomenon, which finds no analog in SISO systems, thus highlights the important role of directionality in sensitivity tradeoff and more generally, in the design of MIMO systems.

A more sophisticated and accordingly, more informative variant of Bode integrals is the Poisson integral for sensitivity and complementary sensitivity functions (Freudenberg and Looze 1985), which can be used to provide quantitative estimates of the waterbed effect. MIMO versions of Poisson integrals are also available (Chen 1995, 2000).

Frequency-Domain Performance Bounds

Performance bounds complement the integral relations and provide fundamental thresholds to the best possible performance ever attainable. Such bounds are useful in providing benchmarks for evaluating a system's performance prior to and after controller design. In the frequency domain, fundamental limits can be specified as the minimal peak magnitude of the sensitivity and complementary sensitivity functions achievable by feedback, or formally, the minimal achievable $\mathcal{H}_\infty$ norms:

$$\gamma_{\min}^S := \inf\{\|S(s)\|_\infty : K(s) \text{ stabilizes } P(s)\},$$

$$\gamma_{\min}^T := \inf\{\|T(s)\|_\infty : K(s) \text{ stabilizes } P(s)\}.$$

Drawing upon *Nevanlinna-Pick interpolation* theory for analytic functions, one can obtain exact performance limits under rather general circumstances (Chen 2000).

$\mathcal{H}_\infty$ **Performance Limits** *Let $z_i \in \mathbb{C}_+$ be the nonminimum phaze zeros of $P(s)$ with left direction vectors w_i, and $p_i \in \mathbb{C}_+$ the unstable poles*

of $P(s)$ with right direction vectors η_i, where z_i and p_i are all distinct. Then,

$$\gamma_{\min}^S = \gamma_{\min}^T = \sqrt{1 + \overline{\sigma}^2 \left(Q_p^{-1/2} Q_{zp} Q_z^{-1/2} \right)},$$

where Q_z, Q_p, and Q_{zp} are the matrices given by

$$Q_z := \left[\frac{w_i^H w_j}{z_i + \overline{z}_j} \right], \quad Q_p := \left[\frac{\eta_i^H \eta_j}{\overline{p}_i + p_j} \right],$$

$$Q_{zp} := \left[\frac{w_i^H \eta_j}{z_i - p_j} \right].$$

More explicit bounds showing how zeros and poles may interact to have an effect on these limits can be obtained, e.g., as

$$\gamma_{\min}^S = \gamma_{\min}^T \geq$$
$$\sqrt{\sin^2 \angle(w_i, \eta_j) + \left| \frac{p_j + \overline{z_i}}{p_j - z_i} \right|^2 \cos^2 \angle(w_i, \eta_j)},$$

which demonstrates once again that the pole and zero directions play an important role in MIMO systems. Note that for RHP poles and zeros located in the close vicinity, this bound can become excessively large, which serves as another vindication why unstable pole-zero cancelation must be prohibited. Note also that for MIMO systems however, whether near pole-zero cancelation is problematic depends additionally on the mutual orientation of the pole and zero directions.

Tracking and Regulation Limits

Tracking and regulation are two canonical objectives of servo mechanisms and constitute chief criteria in assessing the performance of feedback control systems. Understandings gained from these problems will shed light into more general issues indicative of feedback design. In its full generality, a tracking system can be depicted as in Fig. 2, in which a 2-DOF (degree of freedom) controller K is to be designed for

the output z to track a given reference input r, based on the feedforward of the reference signal r and the feedback of the measured output y. The tracking performance is defined in the time domain by the integral square error

$$J = \int_0^\infty \|z(t) - r(t)\|_2^2 dt,$$

Typically, we take r to be a step signal, which in the MIMO setting corresponds to a unitary constant vector, i.e., $r(t) = v$, $t > 0$ and $r(t) = 0$, $t < 0$, where $\|v\|_2 = 1$. We assume P to be LTI. But K can be arbitrarily general, as long as it is causal and stabilizing. We want z to not only track r asymptotically but also minimize J. But how small can it be?

For the regulation problem, a general setup is given in Fig. 3. Likewise, K may be taken as a 2-DOF controller. The control output energy is measured by the quadratic cost

$$E = \int_0^\infty \|u(t)\|_2^2 dt.$$

We consider a disturbance signal d, typically taken as an impulse signal, $d(t) = v\delta(t)$, where v is a unitary vector. In this case, the disturbance can be interpreted as a nonzero initial condition, and the controller K is to regulate the system's zero-input response. Similarly, we assume that P
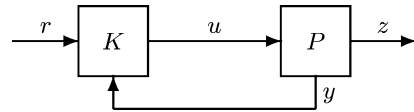
Fundamental Limitation of Feedback Control, Fig. 2
2-DOF tracking control structure
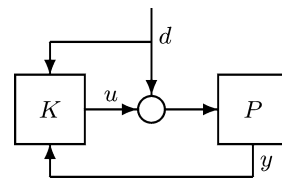
Fundamental Limitation of Feedback Control, Fig. 3
2-DOF regulator

is LTI, but allow K to be any causal, stabilizing controller. Evidently, for a stable P, the problem is trivial; the response will restore itself to the origin and thus no energy is required. But what if the system is unstable? How much energy does the controller must generate to combat the disturbance? What is the smallest amount of energy required? These questions are answered by the best achievable limits of the tracking and regulation performance (Chen et al. 2000, 2003).

Tracking and Regulation Performance Limits
Let $p_i \in \mathbb{C}_+$ and $z_i \in \mathbb{C}_+$ be the RHP poles and zeros of $P(s)$, respectively. Then,

$$\inf\{E : K \text{ stabilizes } P(s)\} = \sum_i p_i \cos^2(\zeta_i, v),$$

$$\inf\{J : K \text{ stabilizes } P(s)\} = \sum_i \frac{1}{z_i} \cos^2(\xi_i, v),$$

where ζ_i and ξ_i are some unitary vectors related to the right pole direction vectors associated with p_i and the left zero direction vectors associated with z_i, respectively.

It becomes instantly clear that the optimal performance depends on both the pole/zero locations and their directions. In particular, it depends on the mutual orientation between the input and pole/zero directions. This sheds some interesting light. Take the tracking performance for an example. For a SISO system, the minimal tracking error can never be made zero for a nonminimum phase plant; in other words, perfect tracking can never be achieved. Yet this is possible for MIMO systems, when the input and zero directions are appropriately aligned, specifically when they are orthogonal. Interestingly, the optimal performance in both cases can be achieved by LTI controllers, though allowed to be more general. As a result, the results herein provide the true fundamental limits that cannot be further improved, in spite of using any other more general forms such as nonlinear, time-varying feedforward and feedback. It is simply the best one can ever hope for, and the LTI controllers turn out to be optimal.

Summary and Future Directions

Whether in time or frequency domain, while the results presented herein may differ in forms and contexts, they unequivocally point to the fact that inherent constraints exist in feedback design, and fundamental limitations will necessarily arise, limiting the performance achievable regardless of controller design. Such constraints and limitations are especially exacerbated by the nonminimum phase zeros and unstable poles in the system. Understanding of these constraints and limitations proves essential to the success of control design.

For both its intrinsic appeal and fundamental implications, the study of fundamental control limitations will continue to be a topic of enduring vitality and indeed will prove timeless. Challenges are especially daunting and endeavor is called for, e.g., to incorporate information and communication constraints into control limitation studies, of which networked control and multi-agent systems serve as notable testimonies.

Cross-References

▶ H-Infinity Control
▶ H$_2$ Optimal Control
▶ Linear Quadratic Optimal Control

Bibliography

Astrom KJ (2000) Limitations on control system performance. Eur J Control 6:2–20

Bode HW (1945) Network analysis and feedback amplifier design. Van Nostrand, Princeton

Chen J (1995) Sensitivity integral relations and design tradeoffs in linear multivariable feedback systems. IEEE Trans Autom Control 40(10):1700–1716

Chen J (1998) Multivariable gain-phase and sensitivity integral relations and design tradeoffs. IEEE Trans Autom Control 43(3):373–385

Chen J (2000) Logarithmic integrals, interpolation bounds, and performance limitations in MIMO systems. IEEE Trans Autom Control 45:1098–1115

Chen J, Qiu L, Toker O (2000) Limitations on maximal tracking accuracy. IEEE Trans Autom Control 45(2):326–331

Chen J, Hara S, Chen G (2003) Best tracking and regulation performance under control energy constraint. IEEE Trans Autom Control 48(8): 1320–1336

Freudenberg JS, Looze DP (1985) Right half plane zeros and poles and design tradeoffs in feedback systems. IEEE Trans Autom Control 30(6): 555–565

Middleton RH (1991) Trade-offs in linear control system design. Automatica 27(2):281–292

Qiu L, Davison EJ (1993) Performance limitations of non-minimum phase systems in the servomechanism problem. Automatica 29(2):337–349

Seron MM, Braslavsky JH, Goodwin GC (1997) Fundamental limitations in filtering and control. Springer, London

G

Game Theory for Security

Tansu Alpcan
Department of Electrical and Electronic
Engineering, The University of Melbourne,
Melbourne, Australia

Abstract

Game theory provides a mature mathematical foundation for making security decisions in a principled manner. Security games help formalizing security problems and decisions using quantitative models. The resulting analytical frameworks lead to better allocation of limited resources and result in more informed responses to security problems in complex systems and organizations. The game-theoretic approach to security is applicable to a wide variety of systems and critical infrastructures such as electricity, water, financial services, and communication networks.

Keywords

Complex systems; Cyberphysical system security; Game theory; Security games

Introduction

Securing a system involves making numerous decisions whether the system is a computer network, part of a business process in an organization, or belongs to a critical infrastructure. One has to decide on, for example, how to configure sensors for surveillance, collect further information on system properties, allocate resources to secure a critical segment, or who should be able to access a specific function in the system. The decision-maker can be, depending on the setting, a regular employee, a system administrator, or the chief technical officer of an organization. In many cases, the decisions are made automatically by a computer program such as allowing a packet pass the firewall or filtering it out. The time frame of these decisions exhibits a high degree of variability from milliseconds, if made by software, to days and weeks, e.g., when they are part of a strategic plan. Each security decision has a cost and any decision-maker is always constrained by limited amount of available resources. More importantly, each decision carries a security risk that needs to be taken into account when balancing the costs and the benefits.

Security games facilitate building analytical models which capture the interaction between malicious attackers, who aim to compromise networks, and owners or administrators defending them. Attacks exploiting vulnerabilities of the underlying systems and defensive countermeasures constitute the moves of the game. Thus, the strategic struggle between attackers and defenders is formalized quantitatively based on the solid mathematical foundation provided by the field of game theory.

An important aspect of security games is the allocation of limited available resources from the perspectives of both attackers and defenders. If the players had access to unlimited resources (e.g., time, computing power, bandwidth), then the resulting security games would be trivial. In real-world security settings, however, both attackers and defenders have to act strategically and make numerous decisions when allocating their respective resources. Unlike in an optimization approach, security games take into account the decisions and resource limitations of both the attackers and the defenders.
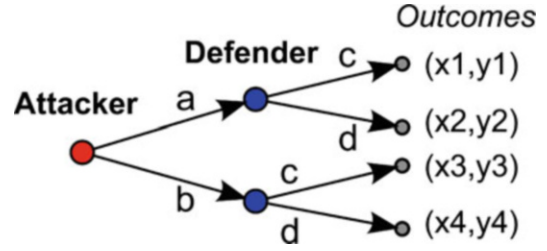
Security Games

A security game is defined with four components: the players, the set of possible actions or strategies for each player, the outcome of the game for each player as a result of their action-reaction, and information structures in the game. The players have their own (selfish or malicious) motivations and inherent resource constraints. Based on these motivations and information available, they choose the most beneficial strategies for themselves and act accordingly. Hence, game theory helps analyzing decision-makers interacting on a system in a quantitative manner.

An Example Formulation

A simple security game can be formulated as a two-player and strategic (noncooperative) one, where one player is the *attacker* and the other one is the *defender* protecting a system. Let the discrete actions available to the attacker and the defender be $\{a, b\}$ and $\{c, d\}$, respectively. Each attack-defense pair leads to one of the outcome pairs for the attacker and the defender $\{(x1, y1), (x2, y2), (x3, y3), (x4, y4)\}$, which represent the respective player's gains (or losses). This security game is depicted graphically in Fig. 1. It can also be represented as a matrix game as follows, where the attacker is the row player and the defender is the column player:

$$
\begin{array}{cc}
(c) & (d) \\
\begin{bmatrix} (x1, y1) & (x2, y2) \\ (x3, y3) & (x4, y4) \end{bmatrix} & \begin{array}{c} (a) \\ (b) \end{array}
\end{array}
$$

Game Theory for Security, Fig. 1 A simple, two-player security game

If the decision variables of the players are continuous, for example, $x \in [0, a]$ and $y \in [0, c]$ denote attack and defense intensity, respectively, then the resulting continuous-kernel game is described using functions instead of a matrix. Then, $J^{\text{attacker}}(x, y)$ and $J^{\text{defender}}(x, y)$ quantify the cost of the attacker and defender as a function of their actions, respectively.

Security Game Types

In its simplest formulation, the conflict between those defending a system and malicious attackers targeting it can be modeled as a two-person zero-sum security game, where the loss of a player is the gain of the other. Alternatively, two- and multi-person nonzero-sum security games generalize this for capturing a broader range of interactions. Static game formulations and their repeated versions are helpful for modeling myopic behavior of players in fast changing situations where planning future actions is of little use. In the case where the underlying system dynamics are predictable and available to the players, dynamic security game formulations can be utilized. If there is an order of actions in the game, for example, a purely reactionary defender, then leader-follower games can be used to formulate such cases where the attacker takes the lead and the defender follows.

Within the framework of security games, the concept of Nash equilibrium, where no player gains from deviating from its own Nash equilibrium strategy if others stick with theirs, provides a solid foundation. However, there are refinements and additional solution concepts when the game is dynamic or when there is more than one

Nash equilibrium or information limitations in the game. These constitute an open and ongoing research topic. A related research question is the design of security games to ensure a favorable outcome from a global perspective while taking into account the independence of individual players in their decisions.

In certain cases, it is useful to analyze the player interactions in multiple layers. For example, in some security games there may be defenders and malicious attackers trying to influence a population of other players by indirect means. In order to circumvent modeling complexity of the problem, evolutionary games have been suggested. While evolutionary games forsake modeling individual player actions, they provide valuable insights to collective behavior of populations of players and ensure tractability. Such models are useful, for example, in the analysis of various security policies affecting many users or security of large-scale critical systems.

Another important aspect of security decisions is the availability and acquisition of information on the properties of the system at hand, the actions of other players, and the incentives behind them. Clearly, the amount of information available has a direct influence on the decisions, yet acquiring information is often costly or even infeasible in some cases. Then, the decisions have to be made with partial information, and information collection becomes part of the decision process itself, creating a complex feedback loop.

Statistical or machine learning techniques and system identification are other useful methods in the analysis of security games where players use the acquired information iteratively to update their own model of the environment and other players. The players then decide on their best courses of action. Existing work on fictitious play and reinforcement learning methods such as Q-learning are applicable and useful. A unique feature of security games is the fact that players try to hide their actions from others. These observability issues and distortions in observations can be captured by modeling the interaction between players who observe each other's actions as a noisy communication channel.

Applications

An early application of the decision and game-theoretic approach has been to the well-defined jamming problem, where malicious attackers aim to disrupt wireless communication between legitimate parties (Kashyap et al. 2004; Zander 1990). Detection of security intrusion and anomalies due to attacks is another problem, where the interaction between attackers and defenders has been modeled successfully using game theory (Alpcan and Başar 2011; Kodialam and Lakshman 2003). Decision and game-theoretic approaches have been applied to a broad variety of networked system security problems such as security investments in organizations (Miura-Ko et al. 2008), (location) privacy (Buttyan and Hubaux 2008; Kantarcioglu et al. 2011), distributed attack detection, attack trees and graphs, adversarial control (Altman et al. 2010), network path selection (Zhang et al. 2010) and topology planning in presence of adversaries (Gueye et al. 2010), as well as to other types of security games and decisions. More recently, security games have been used to investigate cyberphysical security of (smart) power grid (Law et al. 2012). The proceedings of the last three Conferences on Decision and Game Theory for Security published as edited volumes in 2010 (Alpcan et al. 2010), 2011 (Baras et al. 2011), and 2012 (Grossklags and Walrand 2012) as well as the recent survey paper (Manshaei et al. 2013) present an extensive segment of the literature on the subject.

Analytical risk management is a related emerging research subject. Analytical methods and game theory have been applied to the field only recently but with increasing success (Guikema 2009; Mounzer et al. 2010). Another emerging topic is the adversarial mechanism design (Chorppath and Alpcan 2011; Roth 2008), where the goal is to design mechanisms resistant to malicious behavior.

Summary and Future Directions

Game theory provides quantitative methods for studying the players in security problems such

as attackers, defenders, and users as well as their interaction and incentives. Hence, it facilitates making decisions on the best courses of action in addressing security problems while taking into account resource limitations, underlying incentive mechanisms, and security risks. Thus, security games and associated quantitative models have started to replace the prevalent ad hoc decision processes in a wide variety of security problems from safeguarding critical infrastructure to risk management, trust, and privacy in networked systems.

Game theory for security is a young and active research area as evidenced by the recently initiated conference series "Conference on Decision and Game Theory for Security" (www.gamesec-conf.org), the increasing number of journal and conference articles, as well as the recently published books (Alpcan and Başar 2011; Buttyan and Hubaux 2008; Tambe 2011).

Cross-References

- ▶ Dynamic Noncooperative Games
- ▶ Evolutionary Games
- ▶ Learning in Games
- ▶ Network Games
- ▶ Stochastic Games and Learning
- ▶ Strategic Form Games and Nash Equilibrium

Bibliography

Alpcan T, Başar T (2011) Network security: a decision and game theoretic approach. Cambridge University Press, Cambridge, UK. http://www.tansu.alpcan.org/book.php

Alpcan T, Buttyan L, Baras J (eds) (2010) Proceedings of the first international conference on decision and game theory for security, GameSec 2010, Berlin, 22–23 Nov 2010. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg. doi:10.1007/978-3-642-17197-0

Altman E, Başar T, Kavitha V (2010) Adversarial control in a delay tolerant network. In: Alpcan T, Buttyán L, Baras J (eds) Decision and game theory for security. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg, pp 87–106. doi:10.1007/978-3-642-17197-0_6

Baras J, Katz J, Altman E (eds) (2011) Proceedings of the second international conference on decision and game theory for security, GameSec 2011, College Park, 14–15 Nov 2011. Lecture notes in computer science, vol 7037. Springer, Berlin/Heidelberg. doi:10.1007/978-3-642-25280-8

Buttyan L, Hubaux JP (2008) Security and cooperation in wireless networks. Cambridge University Press, Cambridge. http://secowinet.epfl.ch

Chorppath AK, Alpcan T (2011) Adversarial behavior in network mechanism design. In: Proceedings of the of 4th international workshop on game theory in communication networks (Gamecomm), ENS, Cachan

Grossklags J, Walrand JC (eds) (2012) Proceedings of the third international conference on decision and game theory for security, GameSec 2012, Budapest, 5–6 Nov 2012. Lecture notes in computer science, vol 7638. Springer

Gueye A, Walrand J, Anantharam V (2010) Design of network topology in an adversarial environment. In: Alpcan T, Buttyán L, Baras J (eds) Decision and game theory for security. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg, pp 1–20. doi:10.1007/978-3-642-17197-0_1

Guikema SD (2009) Game theory models of intelligent actors in reliability analysis: an overview of the state of the art. In: Bier VM, Azaiez MN (eds) Game theoretic risk analysis of security threats, Springer, New York, pp 1–19. doi:10.1007/978-0-387-87767-9

Kantarcioglu M, Bensoussan A, Hoe S (2011) Investment in privacy-preserving technologies under uncertainty. In: Baras J, Katz J, Altman E (eds) Decision and game theory for security. Lecture notes in computer science, vol 7037. Springer, Berlin/Heidelberg, pp 219–238. doi:10.1007/978-3-642-25280-8_17

Kashyap A, Başar T, Srikant R (2004) Correlated jamming on MIMO Gaussian fading channels. IEEE Trans Inf Theory 50(9):2119–2123. doi:10.1109/TIT.2004.833358

Kodialam M, Lakshman TV (2003) Detecting network intrusions via sampling: a game theoretic approach. In: Proceedings of 22nd IEEE conference on computer communications (Infocom), San Fransisco, vol 3, pp 1880–1889

Law YW, Alpcan T, Palaniswami M (2012) Security games for voltage control in smart grid. In: 50th annual Allerton conference on communication, control, and computing (Allerton 2012), Monticello, IL, USA, pp 212–219. doi:10.1109/Allerton.2012.6483220

Manshaei MH, Zhu Q, Alpcan T, Başar T, Hubaux JP (2013) Game theory meets network security and privacy. ACM Comput Surv 45(3):25:1–25:39. doi:10.1145/2480741.2480742

Miura-Ko RA, Yolken B, Bambos N, Mitchell J (2008) Security investment games of interdependent organizations. In: 46th annual Allerton conference, Monticello, IL, USA

Mounzer J, Alpcan T, Bambos N (2010) Dynamic control and mitigation of interdependent IT security risks. In: Proceedings of the IEEE conference on communication (ICC), Cape Town. IEEE Communications Society

Roth A (2008) The price of malice in linear congestion games. In: WINE '08: proceedings of the 4th international workshop on internet and network economics, Shanghai, pp 118–125

Tambe M (2011) Security and game theory: algorithms, deployed systems, lessons learned. Cambridge University Press, New York, NY, USA

Zander J (1990) Jamming games in slotted Aloha packet radio networks. In: IEEE military communications conference (MILCOM), Morleley, vol 2, pp 830–834. doi:10.1109/MILCOM.1990.117531

Zhang N, Yu W, Fu X, Das S (2010) gPath: a game-theoretic path selection algorithm to protect Tor's anonymity. In: Alpcan T, Buttyán L, Baras J (eds) Decision and game theory for security. Lecture notes in computer science, vol 6442. Springer, Berlin/Heidelberg, pp 58–71. doi:10.1007/978-3-642-17197-0_4

Game Theory: Historical Overview

Tamer Başar
Coordinated Science Laboratory, University of Illinois, Urbana, IL, USA

Abstract

This article provides an overview of the aspects of game theory that are covered in this *Encyclopedia*, which includes a broad spectrum of topics on static and dynamic game theory. It starts with a brief overview of game theory, identifying its basic ingredients, and continues with a brief historical account of the development and evolution of the field. It concludes by providing pointers to other articles in the *Encyclopedia* on game theory, and a list of references.

Keywords

What Is Game Theory?

Game theory deals with strategic interactions among multiple decision makers, called *players* (and in some context *agents*), with each player's preference ordering among multiple alternatives captured in an objective function for that player, which she either tries to maximize (in which case the objective function is a *utility* function or a *benefit* function) or minimize (in which case we refer to the objective function as a *cost* function or a *loss* function). For a nontrivial game, the objective function of a player depends on the choices (*actions* or equivalently *decision variables*) of at least one other player, and generally of all the players, and hence a player cannot simply optimize her own objective function independent of the choices of the other players. This thus brings in a coupling among the actions of the players and binds them together in decision making even in a noncooperative environment. If the players are able to enter into a cooperative agreement so that the selection of actions or decisions is done collectively and with full trust, so that all players would benefit to the extent possible, then we would be in the realm of *cooperative game theory*, where issues such as bargaining and characterization of *fair* outcomes, coalition formation, and excess utility distribution are of relevance; an article in this *Encyclopedia* (by Haurie) discusses cooperation and cooperative outcomes in the context of dynamic games. Other aspects of cooperative game theory can be found in several standard texts on game theory, such as Owen (1995), Vorob'ev (1977), or Fudenberg and Tirole (1991). See also the 2009 survey article Saad et al. (2009), which emphasizes applications of cooperative game theory to communication networks.

If no cooperation is allowed among the players, then we are in the realm of *noncooperative game theory*, where first one has to introduce a satisfactory solution concept. Leaving aside for the moment the issue of how the players can

reach such a solution point, let us address the issue of what would be the minimum features one would expect to see there. To first order, such a solution point should have the property that if all players but one stay put, then the player who has the option of moving away from the solution point should not have any incentive to do so because she cannot improve her payoff. Note that we cannot allow two or more players to move collectively from the solution point, because such a collective move requires cooperation, which is not allowed in a noncooperative game. Such a solution point where none of the players can improve her payoff by a unilateral move is known as a *noncooperative equilibrium* or *Nash equilibrium*, named after John Nash, who introduced it and proved that it exists in finite games (i.e., games where each player has only a finite number of alternatives), over 60 years ago; see Nash (1950, 1951). This result and its various extensions for different frameworks as well as its computation (both off-line and online) are discussed in several articles in this *Encyclopedia*. Another noncooperative equilibrium solution concept is the *Stackelberg equilibrium*, introduced in von Stackelberg (1934), and predating the Nash equilibrium, where there is a hierarchy in decision making among the players, with some of the players, designated as *leaders*, having the ability to first announce their actions (and make a commitment to play them) and the remaining players, designated as *followers*, taking these actions as given in the process of computation of their noncooperative (Nash) equilibria (among themselves). Before announcing their actions, the leaders would of course anticipate these responses and determine their actions in a way that the final outcome will be most favorable to them (in terms of their objective functions). For a comprehensive treatment of Nash and Stackelberg equilibria for different classes of games, see Başar and Olsder (1999).

We say that a noncooperative game is *nonzero-sum* if the sum of the players' objective functions cannot be made zero after appropriate positive scaling and/or translation that do not depend on the players' decision variables. We say that a two-player game is *zero-sum* if the sum of the objective functions of the two players is *zero* or can be made zero by appropriate positive scaling and/or translation that do not depend on the decision variables of the players; hence, two-player zero-sum games can be viewed as a special subclass of two-player nonzero-sum games, and in this case the Nash equilibrium becomes the *saddle-point equilibrium*. A game is a *finite game* if each player has only a finite number of alternatives, that is, the players pick their actions out of finite sets (action sets); otherwise, the game is an *infinite game*. Finite games are also known as *matrix games*. An infinite game is said to be a *continuous-kernel game* if the action sets of the players are continua and the players' objective functions are continuous with respect to action variables of all players. A game is said to be *deterministic* if the players' actions uniquely determine the outcome, as captured in the objective functions, whereas if the objective function of at least one player depends on an additional variable (state of nature) with a probability distribution known to all players (or can be learned on line), then we have a *stochastic game*. A game is a *complete information* game if the description of the game (i.e., the players, the objective functions, and the underlying probability distributions (if stochastic)) is common information to all players; otherwise, we have an *incomplete information* game. We say that a game is *static* if players have access to only the a priori information (shared by all) and none of the players has access to information on the actions of any of the other players; otherwise, what we have is a *dynamic game*. A game is a *single-act game* if every player acts only once; otherwise, the game is *multi-act*. Note that it is possible for a single-act game to be dynamic and for a multi-act game to be static. A dynamic game is said to be a *differential game* if the evolution of the decision process (controlled by the players over time) takes place in continuous time and generally involves a differential equation; if it takes place over a discrete-time horizon, the dynamic game is sometimes called a *discrete-time game*.

In dynamic games, as the game progresses players acquire information (complete or partial)

on past actions of other players and use this information in selecting their own actions (also dictated by the equilibrium solution concept at hand). In finite dynamic games, for example, the progression of a game involves a *tree structure* (also called *extensive form*) where each node is identified with a player along with the time when she acts, and branches emanating from a node show the possible moves of that particular player. A player, at any point in time, could generally be at more than one node, which is a situation that arises when the player does not have complete information on the past moves of other players and hence may not know with certainty which particular node she is at at any particular time. This uncertainty leads to a clustering of nodes into what is called *information sets* for that player. What players decide on within the framework of the extensive form is not their actions, but their *strategies*, that is, what action they would take at each information set (in other words, correspondences between their information sets and their allowable actions). They then take specific actions (or actions are executed on their behalf), dictated by the strategies chosen as well as the progression of the game (decision) process along the tree. The equilibrium is then defined in terms of not actions but strategies.

The notion of a *strategy*, as a mapping from the collection of information sets to action sets, extends readily to infinite dynamic games, and hence, in both differential games and difference games, Nash equilibria are defined in terms of strategies. Several articles in this *Encyclopedia* discuss such equilibria, for both zero-sum and nonzero-sum dynamic games, with and without the presence of probabilistic uncertainty.

In the broad scheme of things, game theory and particularly noncooperative game theory can be viewed as an extension of two fields, both covered in this *Encyclopedia*: *Mathematical Programming* and *Optimal Control Theory*. Any problem in game theory collapses to a problem in one of these disciplines if there is only one player. One-player static games are essentially mathematical programming problems (linear programming or nonlinear programming), and

one-player difference or differential games can be viewed as optimal control problems.

Highlights on the History and Evolution of Game Theory

Game theory has enjoyed over 70 years of scientific development, with the publication of the *Theory of Games and Economic Behavior* by von Neumann and Morgenstern (1947) generally acknowledged to *kick-start* the field. It has experienced incessant growth in both the number of theoretical results and the scope and variety of applications. As a recognition of the vitality of the field, through 2012 a total of 10 Nobel Prizes were given in Economic Sciences for work primarily in game theory, with the first such recognition bestowed in 1994 on John Harsanyi, John Nash, and Reinhard Selten "for their pioneering analysis of equilibria in the theory of noncooperative games." The second round of Nobel Prizes in game theory went to Robert Aumann and Thomas Schelling in 2005, "for having enhanced our understanding of conflict and cooperation through game-theory analysis." The third round recognized Leonid Hurwicz, Eric Maskin, and Roger Myerson in 2007, "for having laid the foundations of mechanism design theory." And the most recent one was in 2012, recognizing Alvin Roth and Lloyd Shapley, "for the theory of stable allocations and the practice of market design." To this list of highest-level awards related to contributions to game theory, one should also add the 1999 Crafoord Prize (which is the highest prize in Biological Sciences), which went to John Maynard Smith (along with Ernst Mayr and G. Williams) "for developing the concept of evolutionary biology," where Smith's recognized contributions had a strong game-theoretic underpinning, through his work on evolutionary games and evolutionary stable equilibrium (Smith 1974, 1982; Smith and Price 1973); this is the topic of one of the articles in this *Encyclopedia* (by Altman). Several other "game theory" articles in the *Encyclopedia* also relate to the

contributions of the Nobel Laureates mentioned above.

Even though von Neumann and Morgenstern's 1944 book is taken as the starting point of the scientific approach to game theory, game-theoretic notions and some isolated key results date back to earlier years and even centuries. Sixteen years earlier, in 1928, von Neumann himself had resolved completely an open fundamental problem in zero-sum games, that *every finite two-player zero-sum game admits a saddle point in mixed strategies*, which is known as the *Minimax Theorem* (von Neumann 1928) – a result which Emile Borel had conjectured to be false eight years before. Some early traces of game-theoretic thinking can be seen in the 1802 work (*Considérations sur la théorie mathématique du jeu*) of André-Marie Ampère (1775–1836), who was influenced by the 1777 writings (Essai d'Arithmétique Morale) of Georges Louis Buffon (1707–1788).

Which event or writing has really started game-theoretic thinking or approach to decision making (in law, politics, economics, operations research, engineering, etc.) may be a topic of debate, but what is indisputable is that in (zero-sum) differential games (which is most relevant to control theory) the starting point was the work of Rufus Isaacs in the RAND Corporation in the early 1950s, which remained classified for at least a decade, before being made accessible to a broad readership in 1965 (Isaacs 1965); see also the review (Ho 1965) which first introduced the book to the control community. One of the articles in this *Encyclopedia* (by Bernhard) talks about this history and the theory developed by Isaacs, within the context of pursuit-evasion games, and another article (again by Bernhard) discusses the impact the zero-sum differential game framework has made on robust control design (Başar and Bernhard 1995). Extension of the game-theoretic framework to nonzero-sum differential games with Nash equilibrium as the solution concept was initiated in Starr and Ho (1969) and with Stackelberg equilibrium as the solution concept in Simaan and Cruz (1973). Systematic study of the role information structures play in the existence of such equilibria

and their uniqueness or nonuniqueness (termed *informational nonuniqueness*) was carried out in Başar (1974, 1976, 1977).

Related Articles on Game Theory in the Encyclopedia

Several articles in the *Encyclopedia* introduce various subareas of game theory and discuss important developments (past and present) in each corresponding area.

The article ▶ Strategic Form Games and Nash Equilibrium introduces the static game framework along with the Nash equilibrium concept, for both finite and infinite games, and discusses the issues of existence and uniqueness as well efficiency. The article ▶ Dynamic Noncooperative Games focuses on dynamic games, again for both finite and infinite games, and discusses extensive form descriptions of the underlying dynamic decision process, either as trees (in finite games) or difference equations (in discrete-time infinite games). Bernhard, in two articles, discusses continuous-time dynamic games, described by differential equations (so-called differential games), but in the two-person zero-sum case. One of these articles ▶ Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games describes the framework initiated by Isaacs, and several of its extensions for pursuit-evasion games, and the other one ▶ Linear Quadratic Zero-Sum Two-Person Differential Games presents results on the special case of linear quadratic differential games, with an important application of that framework to robust control and more precisely H^∞-optimal control.

When the number of players in a nonzero-sum game is countably infinite, or even just sufficiently large, some simplifications arise in the computation and characterization of Nash equilibria. The mathematical framework applicable to this context is provided by *mean field theory*, which is the topic of the article ▶ Mean Field Games, which discusses this relatively new theory within the context of stochastic differential games.

Cooperative solution concepts for dynamic games are discussed in the article ▶ Cooperative Solutions to Dynamic Games, which introduces Pareto optimality, the bargaining solution concept by Nash, characteristic functions, core, and C-optimality, and presents some selected results using these concepts. In the article ▶ Evolutionary Games, the foundations of, as well as the recent advances in, evolutionary games are presented, along with examples showing their potential as a tool for capturing and modeling interactions in complex systems.

The article ▶ Learning in Games addresses the online computation of Nash equilibrium through an iterative process which takes into account each player's response to choices made by the remaining players, with built-in learning and adaptation rules; one such scheme that is discussed in the article is the well-known *fictitious play*. Learning is also the topic of the article ▶ Stochastic Games and Learning, which presents a framework and a set of results using the stochastic games formulation introduced by Shapley in the early 1950s.

The article ▶ Network Games shows how game theory plays an important role in modeling interactions between entities on a network, particularly communication networks, and presents a simple mathematical model to study one such instance, namely, resource allocation in the Internet. How to design a game so as to obtain a desired outcome (as captured by say a Nash equilibrium) is a question central to mechanism design, which is covered in the article ▶ Mechanism Design, which discusses as a specific example the Vickrey-Clarke-Groves (VCG) mechanism.

Two other applications of game theory are to design of auctions and security. The article ▶ Auctions addresses the former, discussing general auction theory along with equilibrium strategies and more specifically combinatorial auctions. The latter is addressed in the article ▶ Game Theory for Security, which discusses how the game-theoretic approach leads to more effective responses to security in complex systems and organizations, with applications to a wide variety of systems and critical infrastructures such as electricity, water, financial services, and communication networks.

Future of Game Theory

The second half of the twentieth century was a golden era for game theory, and all evidence so far in the twenty-first century indicates that the next half century is destined to be a *platinum* era. In all respects game theory is on an upward slope in terms of its vitality, the wealth of topics that fall within its scope, the richness of the conceptual framework it offers, the range of applications, and the challenges it presents to an inquisitive mind.

Cross-References

▶ Auctions
▶ Cooperative Solutions to Dynamic Games
▶ Dynamic Noncooperative Games
▶ Evolutionary Games
▶ Game Theory for Security
▶ Game Theory: Historical Overview
▶ Learning in Games
▶ Linear Quadratic Zero-Sum Two-Person Differential Games
▶ Mean Field Games
▶ Mechanism Design
▶ Network Games
▶ Optimal Control and the Dynamic Programming Principle
▶ Optimization Based Robust Control
▶ Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games
▶ Stochastic Games and Learning
▶ Strategic Form Games and Nash Equilibrium

Bibliography

Başar T (1974) A counter example in linear-quadratic games: existence of non-linear Nash solutions. J Optim Theory Appl 14(4):425–430

Başar T (1976) On the uniqueness of the Nash solution in linear-quadratic differential games. Int J Game Theory 5:65–90

Başar T (1977) Informationally nonunique equilibrium solutions in differential games. SIAM J Control 15(4):636–660

Başar T, Bernhard P (1995) H^∞ optimal control and related minimax design problems: a dynamic game approach, 2nd edn. Birkhäuser, Boston

Başar T, Olsder GJ (1999) Dynamic noncooperative game theory. Classics in applied mathematics, 2nd edn.

SIAM, Philadelphia. (First edition, Academic, London, 1982)

Fudenberg D, Tirole J (1991) Game theory. MIT, Cambridge

Ho Y-C (1965) Review of 'Differential Games' by R. Isaacs. IEEE Trans Autom Control AC-10(4):501–503

Isaacs R (1975) Differential games, 2nd edn. Kruger, New York. (First edition: Wiley, New York, 1965)

Nash JF Jr (1950) Equilibrium points in N-person games. Proc Natl Acad Sci 36(1):48–49

Nash JF Jr (1951) Non-cooperative games. Ann Math 54(2):286–295

Owen G (1995), Game theory, 3rd edn. Academic, New York

Saad W, Han Z, Debbah M, Hjorungnes A, Başar T (2009) Coalitional game theory for communication networks [a tutorial]. IEEE Signal Process Mag 26(5):77–97. Special issue on Game Theory

Simaan M, Cruz JB Jr (1973) On the Stackelberg strategy in nonzero sum games. J Optim Theory Appl 11:533–555

Smith JM (1974) The theory of games and the evolution of animal conflicts. J Theor Biol 47: 209–221

Smith JM (1982) Evolution and the theory of games. Cambridge University Press, Cambridge

Smith JM, Price GR (1973) The logic of animal conflict. Nature 246:15–18

Starr AW, Ho Y-C (1969) Nonzero-sum differential games. J Optim Theory Appl 3:184–206

von Neumann J (1928) Zur theorie der Gesellschaftsspiele. Mathematische Annalen 100:295–320

von Neumann J, Morgenstern O (1947) Theory of games and economic behavior, 2nd edn. Princeton University Press, Princeton. (First edition: 1944)

von Stackelberg H (1934) Marktform und Gleichgewicht, Springer, Vienna. (An English translation appeared in 1952 entitled "The theory of the market economy," published by Oxford University Press, Oxford)

Vorob'ev NH (1977) Game theory. Springer, Berlin

Generalized Finite-Horizon Linear-Quadratic Optimal Control

Augusto Ferrante[1] and Lorenzo Ntogramatzidis[2]
[1]Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy
[2]Department of Mathematics and Statistics, Curtin University, Perth, WA, Australia

Abstract

The linear-quadratic (LQ) problem is the prototype of a large number of optimal control problems, including the fixed endpoint, the point-to-point, and several H_2/H_∞ control problems, as well as the dual counterparts. In the past 50 years, these problems have been addressed using different techniques, each tailored to their specific structure. It is only in the last 10 years that it was recognized that a unifying framework is available. This framework hinges on formulae that parameterize the solutions of the Hamiltonian differential equation in the continuous-time case and the solutions of the extended symplectic system in the discrete-time case. Whereas traditional techniques involve the solutions of Riccati differential or difference equations, the formulae used here to solve the finite-horizon LQ control problem only rely on solutions of the algebraic Riccati equations. In this article, aspects of the framework are described within a discrete-time context.

Keywords

Cyclic boundary conditions; Discrete-time linear systems; Fixed end-point; Initial value; Point-to-point boundary conditions; Quadratic cost; Riccati equations

Introduction

Ever since the *linear-quadratic* (LQ) optimal control problem was introduced in the 1960s by Kalman in his pioneering paper (1960), it has found countless applications in areas such as chemical process control, aeronautics, robotics, servomechanisms, and motor control, to name but a few.

For details on the *raisons d'être* of LQ problems, readers are referred to the classical textbooks on this topic (Anderson and Moore 1971; Kwakernaak and Sivan 1972) and to the Special Issue on LQ optimal control problems in *IEEE Trans. Aut. Contr.*, vol. AC-16, no. 6, 1971. The LQ regulator is not only important per se. It is also the prototype of a variety of fundamental optimization problems. Indeed, several optimal control problems that are extremely relevant in

practice can be recast into composite LQ, dual LQ, or generalized LQ problems. Examples include LQG, H_2 and H_∞ problems, and Kalman filtering problems. Moreover, LQ optimal control is intimately related, via matrix Riccati equations, to absolute stability, dissipative networks, and optimal filtering. The importance of LQ problems is not restricted to linear systems. For example, LQ control techniques can be used to modify an optimal control law in response to perturbations in the dynamics of a nonlinear plant. For these reasons, the LQ problem is universally regarded as a cornerstone of modern control theory.

In its simplest and most classical version, the finite-horizon discrete LQ optimal control can be stated as follows:

Problem 1 Let $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, and consider the linear system

$$x_{t+1} = A x_t + B u_t, \quad y_t = C x_t + D u_t, \quad (1)$$

where the initial state $x_0 \in \mathbb{R}^n$ is given. Let $W = W^\top \in \mathbb{R}^{n \times n}$ be positive semidefinite. Find a sequence of inputs u_t, with $t = 0, 1, \ldots, N-1$, minimizing the cost function

$$J_{N,x_0}(u) \overset{\text{def}}{=} \sum_{t=0}^{N-1} \|y_t\|^2 + x_N^\top W x_N. \quad (2)$$

Historically, LQ problems were first introduced and solved by Kalman in (1960). In this paper, Kalman showed that the LQ problem can be solved for *any* initial state x_0, and the optimal control can be written as a state feedback $u(t) = K(t) x(t)$, where $K(t)$ can be found by solving a famous quadratic matrix difference equation known as the *Riccati equation*. When W is no longer assumed to be positive semidefinite, the optimal solution may or may not exist. A complete analysis of this case has been worked out in Bilardi and Ferrante (2007). In the infinite-horizon case (i.e., when N is infinite), the optimal control (when it exists) is stationary and may be computed by solving an *algebraic* Riccati equation (Anderson and Moore 1971; Kwakernaak and Sivan 1972).

Since its introduction, the original formulation of the classic LQ optimal control problem has been generalized in several different directions, to accommodate for the need of considering more general scenarios than the one represented by Problem 1. Examples include the so-called fixed endpoint LQ, in which the extreme states are sharply assigned, and the *point-to-point* case, in which the initial and terminal values of an output of the system are constrained to be equal to specified values. This led to a number of contributions in the area where different adaptations of the Riccati theory were tailored to these diversified contexts of LQ optimal control. These variations of the classic LQ problem are becoming increasingly important due to their use in several applications of interest. Indeed, many applications including spacecraft, aircraft, and chemical processes involve maneuvering between two states during some phases of a typical mission. Another interesting example is the H_2-optimization of transients in switching plants, where the problem can be divided into a set of finite-horizon LQ problems with welding conditions on the optimal arcs for each switch instant. This problem has been the object of a large number of contributions in the recent literature, under different names: Parameter varying systems, jump linear systems, switching systems, and bumpless systems are definitions extensively used to denote different classes of systems affected by sensible changes in their parameters or structures (Balas and Bokor 2004). In recent years, a new unified approach emerged in Ferrante et al. (2005), Ferrante and Ntogramatzidis (2005), Ferrante and Ntogramatzidis (2007a), and Ferrante and Ntogramatzidis (2007b) that solves the finite-horizon LQ optimal control problem via a formula which parameterizes the set of trajectories generated by the corresponding Hamiltonian differential equation in the continuous time and the extended symplectic difference equation in the discrete case. Loosely, we can say that the expressions parameterizing the trajectories of the Hamiltonian differential equation and the extended symplectic difference equation using this approach hinge on a pair of

"opposite" solutions of the associated algebraic Riccati equations. This active stream of research considerably enlarged the range of optimal control problems that can be successfully addressed. This point of view always requires some controllability-type assumption and the extended symplectic pencil (Ferrante and Ntogramatzidis 2005, 2007b) to be regular and devoid of generalized eigenvalues on the unit circle. More recently, a new point of view has emerged which yields a more direct solution to this problem, without requiring system-theoretic assumptions (Ferrante and Ntogramatzidis 2013a,b; Ntogramatzidis and Ferrante 2013).

The discussion here is restricted to the discrete-time case; for the corresponding continuous-time counterpart, we will only make some comments and refer to the literature.

Notation. For the reader's convenience, we briefly review some, mostly standard, matrix notation used throughout the paper. Given a matrix $B \in \mathbb{R}^{n \times m}$, we denote by $B^\top$ its *transpose* and by $B^\dagger$ its *Moore-Penrose pseudo-inverse*, the unique matrix $B^\dagger$ that satisfies $BB^\dagger B = B$, $B^\dagger B B^\dagger = B^\dagger$, $(BB^\dagger)^\top = BB^\dagger$, and $(B^\dagger B)^\top = B^\dagger B$. The *kernel* of B is the subspace $\{x \in \mathbb{R}^n \mid Bx = 0\}$ and is denoted $\ker B$. The *image* of B is the subspace $\{y \in \mathbb{R}^m \mid \exists x \in \mathbb{R}^n : y = Ax\}$ and is denoted by $\operatorname{im} B$. Given a square matrix A, we denote by $\sigma(A)$ its *spectrum*, i.e., the set of its eigenvalues. We write $A_1 > A_2$ (resp. $A_1 \geq A_2$) when $A_1 - A_2$ is *positive definite* (resp. *positive semidefinite*).

Classical Finite-Horizon Linear-Quadratic Optimal Control

The simplest classical version of the finite-horizon LQ optimal control is Problem 1. By employing some standard linear algebra, this problem may be solved by the classical technique known as "completion of squares": First of all, the cost can be rewritten as

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^\top \ u_t^\top] \Pi \begin{bmatrix} x_t \\ u_t \end{bmatrix}$$

$$+ x_N^\top W x_N, \ \Pi \stackrel{\text{def}}{=} \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix}$$

$$\stackrel{\text{def}}{=} \begin{bmatrix} C^\top \\ D^\top \end{bmatrix} [C \ D] = \Pi^\top \geq 0. \quad (3)$$

Now, let $X_0, X_1, \ldots, X_N$ be an arbitrary sequence of $n \times n$ symmetric matrices. We have the identity

$$\sum_{t=0}^{N-1} \left[x_{t+1}^\top X_{t+1} x_{t+1} - x_t^\top X_t x_t \right]$$
$$+ x_0^\top X_0 x_0 - x_N^\top X_N x_N = 0. \quad (4)$$

Adding (4)–(3) and using the expression (1) for x_{t+1}, we get

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^\top \ u_t^\top]$$

$$\begin{bmatrix} Q + A^\top X_{t+1} A - X_t & S + A^\top X_{t+1} B \\ S^\top + B^\top X_{t+1} A & R + B^\top X_{t+1} B \end{bmatrix}$$

$$\begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_N^\top (W - X_N) x_N + x_0^\top X_0 x_0, \quad (5)$$

which holds for any sequence of matrices X_t. With $X_N \stackrel{\text{def}}{=} W$ fixed, for $t = N-1, N-2, \ldots, 0$, let

$$X_t \stackrel{\text{def}}{=} Q + A^\top X_{t+1} A - (S + A^\top X_{t+1} B)$$

$$(R + B^\top X_{t+1} B)^\dagger (S^\top + B^\top X_{t+1} A). \quad (6)$$

It is now easy to see that all the matrices of the sequence X_t defined above are positive semidefinite. Indeed, $X_N = W \geq 0$. Assume by induction that $X_{t+1} \geq 0$. Then,

$$M_{t+1} \stackrel{\text{def}}{=} \begin{bmatrix} Q + A^\top X_{t+1} A & S + A^\top X_{t+1} B \\ S^\top + B^\top X_{t+1} A & R + B^\top X_{t+1} B \end{bmatrix}$$

$$= \Pi + \begin{bmatrix} A^\top \\ B^\top \end{bmatrix} X_{t+1} [A \ B] \geq 0.$$

Since X_t is the generalized Schur complement of the right upper block of M_{t+1} in M_{t+1}, it follows that $X_t \geq 0$, which in turns implies

$$R + B^\top X_t B \geq 0. \qquad (10)$$

$$J^* = x_0^\top X_0 x_0. \qquad (10)$$

Moreover, by employing Eq. (6) and recalling that, given a positive semidefinite matrix $\Pi_0 = \begin{bmatrix} Q_0 & S_0 \\ S_0^\top & R_0 \end{bmatrix} = \Pi_0^\top \geq 0$, we have $\begin{bmatrix} S_0 R_0^\dagger S_0^\top & S_0 \\ S_0^\top & R_0 \end{bmatrix} = \begin{bmatrix} S_0 \\ R_0 \end{bmatrix} R_0^\dagger [S_0^\top \ \ R_0] \geq 0$, we easily see that

$$M_{t+1} - \begin{bmatrix} X_t & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} S + A^\top X_{t+1} B \\ R + B^\top X_{t+1} B \end{bmatrix} (R + B^\top X_t B)^\dagger$$

$$\begin{bmatrix} S^\top + A^\top X_{t+1} B & R + B^\top X_{t+1} B \end{bmatrix} \geq 0.$$

Hence, (5) takes the form

$$J_{N,x_0}(u) = \sum_{t=0}^{N-1} \| [(R + B^\top X_{t+1} B)^{1/2}]^\dagger$$

$$(S^\top + B^\top X_{t+1} A) x_t + (R + B^\top X_{t+1} B)^{1/2}$$

$$u_t \|_2^2 + x_0^\top X_0 x_0. \qquad (7)$$

Now it is clear that u_t is optimal if and only if

$$(R + B^\top X_{t+1} B)^{1/2}]^\dagger (S^\top + B^\top X_{t+1} A) x_t$$

$$+ (R + B^\top X_{t+1} B)^{1/2} u_t = 0,$$

whose solutions are parameterized by the feedback control

$$u_t = -K_t x_t + G_t v_t, \qquad (8)$$

where $K_t \stackrel{\text{def}}{=} (R + B^\top X_{t+1} B)^\dagger (S^\top + B^\top X_{t+1} A)$ and $G_t \stackrel{\text{def}}{=} [I - (R + B^\top X_{t+1} B)^\dagger (R + B^\top X_{t+1} B)]$ is the orthogonal projector onto the linear space of vectors that can be added to the optimal control u_t without affecting optimality and v_t is a free parameter. The optimal state trajectory is now given by the closed-loop dynamics

$$x_{t+1} = (A - BK_t) x_t + BG_t v_t. \qquad (9)$$

The optimal cost is clearly

The corresponding results in continuous time can be obtained along the same lines as the discrete-time case; see Ferrante and Ntogramatzidis (2013b) and references therein.

More General Linear-Quadratic Problems

The problem discussed in the previous section presents some limitations that prevent its applicability in several important situations. In particular, three relevant generalizations of the classical problem are:

1. *The fixed endpoint case*, where the states at the endpoints x_0 and x_N are both assigned.
2. *The point-to-point case*, where the initial and terminal values z_0 and z_N of linear combination $z_t = V x_t$ of the state of the dynamical system described by (1) are constrained to be equal to two assigned vectors.
3. *The cyclic case*, where the states at the endpoints x_0 and x_N are not sharply assigned, but they are constrained to be equal (clearly, we can have combinations of (2) and (3)).

All these problems are special cases of a general LQ problem that can be stated as follows:

Problem 2 Consider the dynamical setting (1) of Problem 1. Find a sequence of inputs u_t, with $t = 0, 1, \ldots, N - 1$ and an initial state x_0 minimizing the cost function

$$J_{N,x_0}(u) \stackrel{\text{def}}{=} \sum_{t=0}^{N-1} \| y_t \|^2 + [x_0^\top - \overline{x}_0^\top \ \ x_N^\top - \overline{x}_N^\top]$$

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{12}^\top & W_{22} \end{bmatrix} \begin{bmatrix} x_0 - \overline{x}_0 \\ x_N - \overline{x}_N \end{bmatrix} \qquad (11)$$

under the dynamic constraints (1) and the endpoints constraints

$$V \begin{bmatrix} x_0 \\ x_N \end{bmatrix} = v. \qquad (12)$$

Here $W \stackrel{\text{def}}{=} \begin{bmatrix} W_{11} & W_{12} \\ W_{12}^{\top} & W_{22} \end{bmatrix}$ is a positive semidefinite matrix (partitioned in four blocks) that quadratically penalizes the differences between the initial state x_0 and a desired initial state $\overline{x}_0$ and between the final state x_N and a desired final state $\overline{x}_N$ (This general problem formulation can also encompass problems where the difference $x_0 - x_N$ is not fixed but has to be quadratically penalized with a matrix $\Delta = \Delta^{\top} \geq 0$ in the performance index $J'_{N,x_0}(u) = \sum_{t=0}^{N-1} [x_t^{\top} \ u_t^{\top}] \Pi \begin{bmatrix} x_t \\ u_t \end{bmatrix} + (x_0 - x_N)^{\top} \Delta (x_0 - x_N)$. It is simple to see that this performance index can be brought back to (11) by setting $W = \begin{bmatrix} \Delta & -\Delta \\ -\Delta & \Delta \end{bmatrix}$ and $\overline{x}_0 = \overline{x}_N = 0$.). Equation (12) permits to impose also a hard constraint on an arbitrary linear combination of initial and final states.

The solution of Problem 2 can be obtained by parameterizing the solutions of the so-called extended symplectic system (see Ferrante and Levy (1998) and references therein for a discussion on symplectic matrices and pencils). This solution can be convenient also for the classical case of Problem 1. In fact it does not require to iterate the difference Riccati equation (which can be undesirable if the time horizon is large) but only to solve an algebraic Riccati equation and a related discrete Lyapunov equation.

This solution requires some definitions, preliminary results, and standing assumptions (see Ntogramatzidis and Ferrante (2013) and Ferrante and Ntogramatzidis (2013a) for a more general approach which does not require such assumptions). A detailed proof of the main result can be found in Ferrante and Ntogramatzidis (2007b); see also Zattoni (2008) and Ferrante and Ntogramatzidis (2012). The extended symplectic pencil is defined by

$$zF - G, F \stackrel{\text{def}}{=} \begin{bmatrix} I_n & 0 & 0 \\ 0 & -A^{\top} & 0 \\ 0 & -B^{\top} & 0 \end{bmatrix},$$

$$G \stackrel{\text{def}}{=} \begin{bmatrix} A & 0 & B \\ Q & -I_n & S \\ S^{\top} & 0 & R \end{bmatrix}. \tag{13}$$

where Q, S, R are defined as in (3). We make the following assumptions:

(A1) The pair (A, B) is *modulus controllable*, i.e., $\forall \lambda \in \mathbb{C} \setminus \{0\}$ at least one of the two matrices $[\lambda I - A \mid B]$ and $[\lambda^{-1} I - A \mid B]$ has full row rank.

(A2) The pencil $zF - G$ is regular (i.e., $\det(zF - G)$ is not the zero polynomial) and has no generalized eigenvalues on the unit circle.

Consider the discrete algebraic Riccati equation

$$X = A^{\top} X A - (A^{\top} X B + S)(R + B^{\top} X B)^{-1}$$

$$(S^{\top} + B^{\top} X A) + Q. \tag{14}$$

Under assumptions (A1)–(A2), (14) admits a strongly unmixed solution $X = X^{\top}$, i.e., a solution X for which the corresponding closed-loop matrix

$$A_X \stackrel{\text{def}}{=} A - B K_X, \quad K_X \stackrel{\text{def}}{=} (R + B^{\top} X B)^{-1}$$

$$(S^{\top} + B^{\top} X A) \tag{15}$$

has spectrum that does not contain reciprocal values, i.e., $\lambda \in \sigma(A_X)$ implies $\lambda^{-1} \notin \sigma(A_X)$. It can now be proven that the following closed-loop Lyapunov equation admits a unique solution $Y = Y^{\top} \in \mathbb{R}^{n \times n}$:

$$A_X Y A_X^{\top} - Y + B (R + B^{\top} X B)^{-1} B^{\top} = 0. \tag{16}$$

The following theorem provides an explicit formula parameterizing all the optimal state and control trajectories for Problem 2. Notice that this formula can be readily implemented starting from the problem data.

Theorem 1 *With reference to Problem 2, assume that (A1) and (A2) are satisfied. Let $X = X^{\top}$ be any strongly unmixed solution of (14) and $Y = Y^{\top}$ be the corresponding solution of (16). Let N_V be a basis matrix (In the case when $\ker V = \{0\}$, we consider N_V to be void.) of the null space of V. Moreover, let*

$$F \stackrel{\text{def}}{=} A_X^N,$$

$$K_\star \stackrel{\text{def}}{=} K_X\, Y\, A_X^\top - (R + B^\top X\, B)^{-1} B^\top,$$

$$\hat{X} \stackrel{\text{def}}{=} \operatorname{diag}(-X, X), \quad \overline{x} \stackrel{\text{def}}{=} \begin{bmatrix} \overline{x}_0 \\ \overline{x}_N \end{bmatrix},$$

$$w \stackrel{\text{def}}{=} \begin{bmatrix} v \\ -N_V^\top\, W\, \overline{x} \end{bmatrix}, \quad L \stackrel{\text{def}}{=} \begin{bmatrix} I_n & Y\, F^\top \\ F & Y \end{bmatrix},$$

$$U \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -F^\top \\ 0 & I_n \end{bmatrix},$$

$$M \stackrel{\text{def}}{=} \begin{bmatrix} V\, L \\ N_V^\top\, [\, (\hat{X} - W)\, L - U\,] \end{bmatrix}.$$

Problem 2 *admits solutions if and only if* $w \in im\, M$. *In this case, let* N_M *be a basis matrix of the null space of* M, *and define*

$$\mathcal{P} \stackrel{\text{def}}{=} \{\pi = M^\dagger\, w + N_M\, \zeta \mid \zeta \text{ arbitrary}\}. \quad (17)$$

Then, the set of optimal state and control trajectories of Problem 2 *is parameterized in terms of* $\pi \in \mathcal{P}$, *by*

$$\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_X^t & Y\, (A_X^\top)^{N-t} \\ -K_X A_X^t & -K_\star\, (A_X^\top)^{N-t-1} \end{bmatrix} \pi, \\ \qquad\qquad 0 \le t \le N-1, \\ \begin{bmatrix} A_X^N & Y \\ 0 & 0 \end{bmatrix} \pi, \qquad t = N. \end{cases} \quad (18)$$

The interpretation of the above result is the following. As π varies, (18) describes the trajectories of the extended symplectic system. The set $\mathcal{P}$ defined in (17) is the set of π for which these trajectories satisfy the boundary conditions. All the details of this construction can be found in Ferrante and Ntogramatzidis (2007b). If the pair (A, B) is stabilizable, we can choose $X = X^\top$ to be the stabilizing solution of (14). In such case, the matrices A_X^t, $(A_X^\top)^{N-t}$ and $(A_X^\top)^{N-t-1}$ appearing in (18) are asymptotically stable for all $t = 0, \ldots, N$. Thus, in this case, the optimal state trajectory and control are expressed in terms of powers of strictly stable matrices in the overall time interval, thus ensuring the robustness of the obtained solution even for very large time horizons. Indeed, the stabilizing solution of an

algebraic Riccati equation and the solution of a Lyapunov equation may be computed by standard and robust algorithms available in any control package (see the MATLAB® routines `dare.m` and `dlyap.m`). We refer to Ferrante et al. (2005) and Ferrante and Ntogramatzidis (2013b) for the continuous-time counterpart of the above results.

Summary

With the technique discussed in this paper, a large number of LQ problems can be tackled in a unified framework. Moreover, several finite-horizon LQ problems that can be interesting and useful in practice can be recovered as particular cases of the control problem considered here. The generality of the optimal control problem herein considered is crucial in the solution of several $H_2 - H_\infty$ optimization problems whose optimal trajectory is composed of a set of arches, each one solving a parametric LQ subproblem in a specific time horizon and all joined together at the endpoints of each subinterval. In these cases, in fact, a very general form of constraint on the extreme states is essential in order to express the condition of conjunction of each pair of subsequent arches at the endpoints.

Cross-References

▶ H₂ Optimal Control
▶ H-Infinity Control
▶ Linear Quadratic Optimal Control

Bibliography

Anderson BDO, Moore JB (1971) Linear optimal control. Prentice Hall International, Englewood Cliffs

Balas G, Bokor J (2004) Detection filter design for LPV systems – a geometric approach. Automatica 40:511–518

Bilardi G, Ferrante A (2007) The role of terminal cost/reward in finite-horizon discrete-time LQ optimal control. Linear Algebra Appl (Spec Issue honor Paul Fuhrmann) 425:323–344

Ferrante A (2004) On the structure of the solution of discrete-time algebraic Riccati equation with singular

closed-loop matrix. IEEE Trans Autom Control AC-49(11):2049–2054

Ferrante A, Levy B (1998) Canonical form for symplectic matrix pencils. Linear Algebra Appl 274:259–300

Ferrante A, Ntogramatzidis L (2005) Employing the algebraic Riccati equation for a parametrization of the solutions of the finite-horizon LQ problem: the discrete-time case. Syst Control Lett 54(7):693–703

Ferrante A, Ntogramatzidis L (2007a) A unified approach to the finite-horizon linear quadratic optimal control problem. Eur J Control 13(5):473–488

Ferrante A, Ntogramatzidis L (2007b) A unified approach to finite-horizon generalized LQ optimal control problems for discrete-time systems. Linear Algebra Appl (Spec Issue honor Paul Fuhrmann) 425(2–3):242–260

Ferrante A, Ntogramatzidis L (2012) Comments on "Structural Invariant Subspaces of Singular Hamiltonian Systems and Nonrecursive Solutions of Finite-Horizon Optimal Control Problems". IEEE Trans Autom Control 57(1):270–272

Ferrante A, Ntogramatzidis L (2013a) The extended symplectic pencil and the finite-horizon LQ problem with two-sided boundary conditions. IEEE Trans Autom Control 58(8):2102–2107

Ferrante A, Ntogramatzidis L (2013b) The role of the generalised continuous algebraic Riccati equation in impulse-free continuous-time singular LQ optimal control. In: Proceedings of the 52nd conference on decision and control (CDC 13), Florence, 10–13 Dec 2013b

Ferrante A, Marro G, Ntogramatzidis L (2005) A parametrization of the solutions of the finite-horizon LQ problem with general cost and boundary conditions. Automatica 41:1359–1366

Kalman RE (1960) Contributions to the theory of optimal control. Bulletin de la Sociedad Matematica Mexicana 5:102–119

Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York

Ntogramatzidis L, Ferrante A (2010) On the solution of the Riccati differential equation arising from the LQ optimal control problem. Syst Control Lett 59(2):114–121

Ntogramatzidis L, Ferrante A (2013) The generalised discrete algebraic Riccati equation in linear-quadratic optimal control. Automatica 49:471–478. doi:10.1016/j.automatica.2012.11.006

Ntogramatzidis L, Marro G (2005) A parametrization of the solutions of the Hamiltonian system for stabilizable pairs. Int J Control 78(7):530–533

Prattichizzo D, Ntogramatzidis L, Marro G (2008) A new approach to the cheap LQ regulator exploiting the geometric properties of the Hamiltonian system. Automatica 44:2834–2839

Zattoni E (2008) Structural invariant subspaces of singular Hamiltonian systems and nonrecursive solutions of finite-horizon optimal control problems. IEEE Trans Autom Control AC-53(5):1279–1284

Graphs for Modeling Networked Interactions

Mehran Mesbahi[1] and Magnus Egerstedt[2]
[1]University of Washington, Seattle, WA, USA
[2]Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Graphs constitute natural models for networks of interacting agents. This chapter introduces graph theoretic formalisms that facilitate analysis and synthesis of coordinated control algorithms over networks.

Keywords

Distributed control; Graph theory; Multi-agent networks

Introduction

Distributed and networked systems are characterized by a set of dynamical units (agents, actors, nodes) that share information with each other in order to achieve a global performance objective using locally available information. Information can typically be shared if agents are within communication or sensing range of each other. It is useful to abstract away the particulars of the underlying information-exchange mechanism and simply say that an information link exists between two nodes if they can share information. Such an abstraction is naturally represented in terms of a graph.

A graph is a combinatorial object defined by two constructs: vertices (or nodes) and edges (or links) connecting pairs of distinct vertices. The set of N vertices specified by $V = \{v_1, \ldots, v_N\}$ corresponds to the agents, and an edge between vertices v_i and v_j is represented by (v_i, v_j); the set of all edges constitutes the edge set E. The *graph* G is thus the pair $G = (V, E)$ and the interpretation is that an edge $(v_i, v_j) \in E$

Graphs for Modeling Networked Interactions, Fig. 1 A network of agents equipped with omnidirectional range sensors can be viewed as a graph (undirected in this case), with nodes corresponding to the agents and edges to their pairwise interactions, which are enabled whenever the agents are within a certain distance from each other
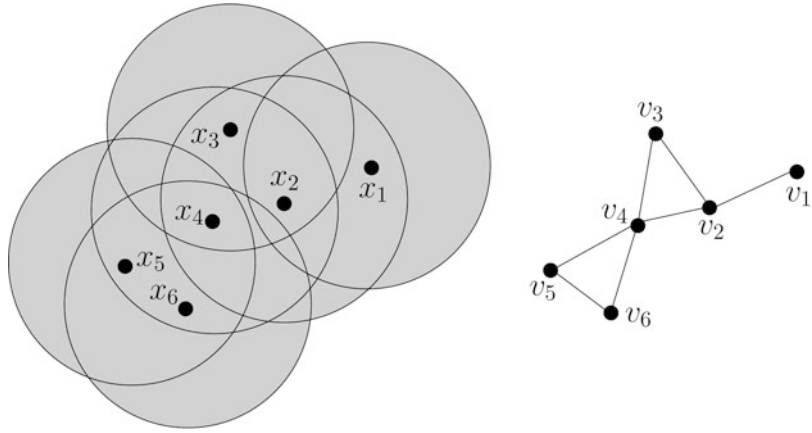
if information can flow from vertex i to vertex j. If the information exchange is sensor based, and if there is a state x_i associated with vertex i, e.g., its position, then this information is typically relative, i.e., the states are measured relative to each other and the information obtained along the edge is $x_j - x_i$. If on the other hand the information is communicated, then the full state information x_i can be transmitted along the edge (v_i, v_j). The graph abstraction for a network of agents with sensor-based information exchange is illustrated in Fig. 1.

It is often useful to differentiate between scenarios where the Information exchange is bidirectional – if agent i can get information from agent j, then agent j can get information from agent i – and when it is not. In the language of graph theory, an undirected graph is one where $(v_i, v_j) \in E$ implies that $(v_j, v_i) \in E$, while a directed graph is one where such an implication may not hold.

Graph-Based Coordination Models

Graphs provide structural insights into how different coordination algorithms behave over a network. A coordination algorithm, or protocol, is an update rule that describes how the node states should evolve over time. To understand such protocols, one needs to connect the interaction dynamics to the underlying graph structure. This connection is facilitated through the common intersection of linear system theory and graph theory, namely, the broad discipline of *algebraic graph theory,* by first associating matrices with graphs. For undirected graphs, the following matrices play a key role:

Degree matrix : $\Delta = \text{Diag}\, (\deg(v_1), \ldots, \deg(v_N))$,

 Adjacency matrix : $A = [a_{ij}]$,

where **Diag** denotes a diagonal matrix whose diagonal consists of its argument and $\deg(v_i)$ is the degree of vertex i in the graph, i.e., the cardinality of the set of edges incident on vertex i. Moreover,

$$a_{ij} = \begin{cases} 1 & \text{if}\,(v_j, v_i) \in E \\ 0 & \text{otherwise.} \end{cases}$$

As an example of how these matrices come into play, the so-called *consensus* protocol over scalar states can be compactly written on ensemble form as

$$\dot{x} = -Lx,$$

where $x = [x_1, \ldots, x_N]^T$ and L is the *graph Laplacian:*

$$L = \Delta - A.$$

A useful matrix for directed networks is the incidence matrix, obtained by associating an index to each edge in E. We say that $v_i = \text{tail}(e_j)$ if edge e_j starts at node v_i and $v_i = \text{head}(e_j)$ if e_j ends up at v_i, leading to the

 Incidence matrix : $D = [\iota_{ij}]$,

where

$$
\iota_{ij} = \begin{cases} 1 & \text{if } v_i = \text{head}(e_j) \\ -1 & \text{if } v_i = \text{tail}(e_j) \\ 0 & \text{otherwise.} \end{cases}
$$

It now follows that for undirected networks, the Laplacian has an equivalent representation as

$$
L = DD^T,
$$

where D is the incidence matrix associated with an arbitrary orientation (assignment of directions to the edges) of the undirected graph. This in turn implies that for undirected networks, L is a positive semi-definite matrix and that all of its eigenvalues are nonnegative.

If the network is directed, one has to pay attention to the direction in which information is flowing, using the *in-degree* and *out-degree* of the vertices. The out-degree of vertex i is the number of directed edges that originate at i, and similarly the in-degree of node i is the number of directed edges that terminate at node i. A directed graph is *balanced* if the out-degree is equal to the in-degree at every vertex in the graph. And, the graph Laplacian for directed graphs is obtained by only counting information flowing in the correct direction, i.e., if $L = [\ell_{ij}]$, then l_{ii} is the in-degree of vertex i and $l_{ij} = -1$ if $i \neq j$ and $(v_j, v_i) \in E$.

As a final note, for both directed and undirected networks, it is possible to associate weights to the edges, $w : E \to \Upsilon$ where Υ is set of nonnegative reals or more generally a field, in which case the Laplacian's diagonal elements are the sum of the weights of edges incident to node i and the off-diagonal elements are $-w(v_j, v_i)$ when $(v_j, v_i) \in E$.

Applications

Graph-based coordination has been used in a number of application domains, such as multi-agent robotics, mobile sensor and communication networks, formation control, and biological systems. One way in which the consensus protocol

can be generalized is by defining an edge-tension energy $E_{ij}(\|x_i - x_j\|)$ along each edge in the graph, which gives the total energy in the network as

$$
E(x) = \sum_{i=1}^{N} \sum_{j \in N_i} E_{ij}(\|x_i - x_j\|).
$$

If the agents update their states in such a way as to reduce the total energy in the system according to a gradient descent scheme, the update law becomes

$$
\dot{x} = -\frac{\partial E(x)}{\partial x_i} \quad \Rightarrow \quad \dot{E}(x) = -\left\| \frac{\partial E(x)}{\partial x} \right\|_2^2,
$$

which is nonpositive, i.e., the total energy is reduced in the network. For undirected networks, the ensemble version of this protocol assumes the form

$$
\dot{x} = -L_w(x)x,
$$

where the weighted graph Laplacian is

$$
L_w(x) = DW(x)D^T,
$$

with the weight matrix $W(x) = \mathbf{Diag}(w_1(x), \ldots, w_M(x))$. Here M is the total number of edges in the network, and $w_k(x)$ is the weight that corresponds to the kth edge, given an arbitrary ordering of the edges consistent with the incident matrix D.

This energy interpretation allows for the synthesis of coordination laws for multi-agent networks with desirable properties, such as $E_{ij}(\|x_i - x_j\|) = (\|x_i - x_j\| - d_{ij})^2$ for making the agents reach the desired interagent distances d_{ij}, as shown in Fig. 2. Other applications where these types of constructions have been used include collision avoidance and connectivity maintenance.

Summary and Future Directions

A number of issues pertaining to graph-based distributed control remain to be resolved. These include how heterogeneous networks, i.e., networks comprising of agents with different capabilities,
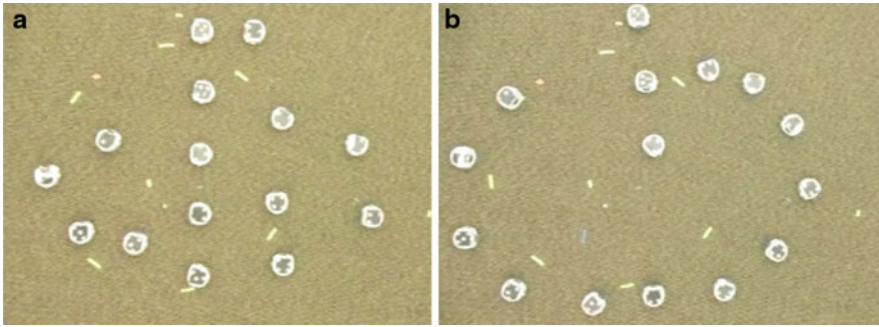
Graphs for Modeling Networked Interactions, Fig. 2 Fifteen mobile robots are forming the letter "G" by executing a weighted version of the consensus protocol. (**a**) Formation control ($t = 0$). (**b**) Formation control ($t = 5$)

can be designed and understood. A variation to this theme is networks of networks, i.e., networks that are loosely coupled together and that must coordinate at a higher level of abstraction. Another key issue concerns how human operators should interact with networked control systems.

Cross-References

- ▶ Averaging Algorithms and Consensus
- ▶ Distributed Optimization
- ▶ Dynamic Graphs, Connectivity of
- ▶ Flocking in Networked Systems
- ▶ Networked Systems
- ▶ Optimal Deployment and Spatial Coverage
- ▶ Oscillator Synchronization
- ▶ Vehicular Chains

Recommended Reading

There are a number of research manuscripts and textbooks that explore the role of network structure on the system theoretic aspects of networked dynamic systems and its many ramifications. Some of these references are listed below.

Bibliography

Bai H, Arcak M, Wen J (2011) Cooperative control design: a systematic, passivity-based approach. Springer, Berlin

Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks: a mathematical approach to motion coordination algorithms. Princeton University Press, Princeton

Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Princeton University Press, Princeton

Ren W, Beard R (2008) Distributed consensus in multi-vehicle cooperative control. Springer, Berlin

G

H

H₂ Optimal Control

Ben M. Chen
Department of Electrical and Computer
Engineering, National University of Singapore,
Singapore, Singapore

Abstract

An optimization-based approach to linear feedback control system design uses the H_2 norm, or energy of the impulse response, to quantify closed-loop performance. In this entry, an overview of state-space methods for solving H_2 optimal control problems via Riccati equations and matrix inequalities is presented in a continuous-time setting. Both regular and singular problems are considered. Connections to so-called LQR and LQG control problems are also described.

Keywords

Feedback control; H_2 control; Linear matrix inequalities; Linear systems; Riccati equations; State-space methods

Introduction

Modern multivariable control theory based on state-space models is able to handle multi-feedback-loop designs, with the added benefit that design methods derived from it are amenable to computer implementation. Indeed, over the last five decades, a number of multivariable analysis and design methods have been developed using the state-space description of systems. Of these design tools, H_2 optimal control problems involve minimizing the H_2 norm of the closed-loop transfer function from exogenous disturbance signals to a pertinent controlled output signals of a given plant by appropriate use of a internally stabilizing feedback controller. It was not until the 1990s that a complete solution to the general H_2 optimal control problem began to emerge. To elaborate on this, let us concentrate our discussion on H_2 optimal control for a continuous-time system Σ expressed in the following state-space form:

$$\dot{x} = Ax + Bu + Ew \tag{1}$$

$$y = C_1 x + D_{11} u + D_1 w \tag{2}$$

$$z = C_2 x + D_2 u + D_{22} w \tag{3}$$

where x is the state variable, u is the control input, w is the exogenous disturbance input, y is the measurement output, and z is the controlled output. The system Σ is typically an augmented or generalized plant model including weighting functions that reflect design requirements. The H_2 optimal control problem is to find an appropriate control law, relating the control input u to the measured output y, such that when it is applied to the given plant in Eqs. (1)–(3), the

resulting closed-loop system is internally stable, and the H_2 norm of the resulting closed-loop transfer matrix from the disturbance input w to the controlled output z, denoted by $T_{zw}(s)$, is minimized. For a stable transfer matrix $T_{zw}(s)$, the H_2 norm is defined as

$$\|T_{zw}\|_2 = \left(\frac{1}{2\pi} \, \text{trace}\left[\int_{-\infty}^{\infty} T_{zw}(j\omega)T_{zw}^{\text{H}}(j\omega)d\omega\right]\right)^{\frac{1}{2}} \tag{4}$$

where T_{zw}^{H} is the conjugate transpose of T_{zw}. Note that the H_2 norm is equal to the energy of the impulse response associated with $T_{zw}(s)$ and this is finite only if the direct feedthrough term of the transfer matrix is zero.

It is standard to make the following assumptions on the problem data: $D_{11} = 0$; $D_{22} = 0$; (A, B) is stabilizable; (A, C_1) is detectable. The last two assumptions are necessary for the existence of an internally stabilizing control law. The first assumption can be made without loss of generality via a constant loop transformation. Finally, either the assumption $D_{22} = 0$ can be achieved by a pre-static feedback law, or the problem does not yield a solution that has finite H_2 closed-loop norm.

There are two main groups into which all H_2 optimal control problems can be divided. The first group, referred to as regular H_2 optimal control problems, consists of those problems for which the given plant satisfies two additional assumptions:

1. The subsystem from the control input to the controlled output, i.e., (A, B, C_2, D_2), has no invariant zeros on the imaginary axis, and its direct feedthrough matrix, D_2, is injective (i.e., it is tall and of full rank).
2. The subsystem from the exogenous disturbance to the measurement output, i.e., (A, E, C_1, D_1), has no invariant zeros on the imaginary axis and its direct feedthrough matrix, D_1, is surjective (i.e., it is fat and of full rank).

Assumption 1 implies that (A, B, C_2, D_2) is left invertible with no infinite zero, and Assumption 2 implies that (A, E, C_1, D_1) is right invertible with no infinite zero. The second, referred to

as singular H_2 optimal control problems, consists of those which are not regular.

Most of the research in the literature was expended on regular problems. Also, most of the available textbooks and review articles, see, for example, Anderson and Moore (1989), Bryson and Ho (1975), Fleming and Rishel (1975), Kailath (1974), Kwakernaak and Sivan (1972), Lewis (1986), and Zhou et al. (1996), to name a few, cover predominantly only a subset of regular problems. The singular H_2 control problem with state feedback was studied in Geerts (1989) and Willems et al. (1986). Using different classes of state- and measurement-feedback control laws, Stoorvogel et al. (1993) studied the general H_2 optimal control problems for the first time. In particular, necessary and sufficient conditions are provided therein for the existence of a solution in the case of state-feedback control, and in the case of measurement-feedback control. Following this, Trentelman and Stoorvogel (1995) explored necessary and sufficient conditions for the existence of an H_2 optimal controller within the context of discrete-time and sampled-data systems. At the same time Chen et al. (1993, 1994a) provided a thorough treatment of the H_2 optimal control problem with state-feedback controllers. This includes a parameterization and construction of the set of all H_2 optimal controllers and the associated sets of H_2 optimal fixed modes and H_2 optimal fixed decoupling zeros. Also, they provided a computationally feasible design algorithm for selecting an H_2 optimal state-feedback controller that places the closed-loop poles at desired locations whenever possible. Furthermore, Chen and Saberi (1993) and Chen et al. (1996) developed the necessary and sufficient conditions for the uniqueness of an H_2 optimal controller. Interested readers are referred to the textbook Saberi et al. (1995) for a detailed treatment of H_2 optimal control problems in their full generality.

Regular Case

Solving regular H_2 optimal control problems is relatively straightforward. In the case that all of

the state variables of the given plant are available for feedback, i.e., $y = x$, and Assumption 1 holds, the corresponding H_2 optimal control problem can be solved in terms of the unique positive semi-definite stabilizing solution $P \geq 0$ of the following algebraic Riccati equation:

$$A^{\mathrm{T}}P + PA + C_2^{\mathrm{T}}C_2 - (PB + C_2^{\mathrm{T}}D_2)(D_2^{\mathrm{T}}D_2)^{-1}$$
$$(D_2^{\mathrm{T}}C_2 + B^{\mathrm{T}}P) = 0 \qquad (5)$$

The H_2 optimal state-feedback law is given by

$$u = Fx = -(D_2^{\mathrm{T}}D_2)^{-1}(D_2^{\mathrm{T}}C_2 + B^{\mathrm{T}}P)\, x \quad (6)$$

and the resulting closed-loop transfer matrix from w to z, $T_{zw}(s)$, has the following property:

$$\|T_{zw}\|_2 = \sqrt{\operatorname{trace}(E^{\mathrm{T}}PE)} \qquad (7)$$

Note that the H_2 optimal state-feedback control law is generally nonunique. A trivial example is the case when $E = 0$, whereby every stabilizing control law is an optimal solution. It is also interesting to note that the closed-loop system comprising the given plant with $y = x$ and the state-feedback control law of Eq. (6) has poles at all the stable invariant zeros and all the mirror images of the unstable invariant zeros of (A, B, C_2, D_2) together with some other fixed locations in the left half complex plane. More detailed results about the optimal fixed modes and fixed decoupling zeros for general H_2 optimal control can be found in Chen et al. (1993).

It can be shown that the well-known linear quadratic regulation (LQR) problem can be reformulated as a regular H_2 optimal control problem. For a given plant

$$\dot{x} = Ax + Bu, \quad x(0) = X_0 \qquad (8)$$

with (A, B) being stabilizable, the LQR problem is to find a control law $u = Fx$ such that the following performance index is minimized:

$$J = \int_0^\infty (x^{\mathrm{T}}Q_\star x + u^{\mathrm{T}}R_\star u)dt, \qquad (9)$$

where $R_\star > 0$ and $Q_\star \geq 0$ with $(A, Q_\star^{\frac{1}{2}})$ being detectable. The LQR problem is equivalent to finding a static state-feedback H_2 optimal control law for the following auxiliary plant Σ_{LQR}:

$$\dot{x} = Ax + Bu + X_0 w \qquad (10)$$

$$y = x \qquad (11)$$

$$z = \begin{pmatrix} 0 \\ Q_\star^{\frac{1}{2}} \end{pmatrix} x + \begin{pmatrix} R_\star^{\frac{1}{2}} \\ 0 \end{pmatrix} u \qquad (12)$$

For the measurement-feedback case with both Assumptions 1 and 2 being satisfied, the corresponding H_2 optimal control problem can be solved by finding a positive semi-definite stabilizing solution $P \geq 0$ for the Riccati equation given in Eq. (5) and a positive semi-definite stabilizing solution $Q \geq 0$ for the following Riccati equation:

$$QA^{\mathrm{T}} + AQ + EE^{\mathrm{T}} - (QC_1^{\mathrm{T}} + ED_1^{\mathrm{T}})(D_1D_1^{\mathrm{T}})^{-1}$$
$$(D_1E^{\mathrm{T}} + C_1Q) = 0 \qquad (13)$$

The H_2 optimal measurement-feedback law is given by

$$\dot{v} = (A + BF + KC_1)v - Ky, \quad u = Fx \quad (14)$$

where F is as given in Eq. (6) and

$$K = -(QC_1^{\mathrm{T}} + ED_1^{\mathrm{T}})(D_1D_1^{\mathrm{T}})^{-1} \qquad (15)$$

In fact, such an optimal control law is unique and the resulting closed-loop transfer matrix from w to z, $T_{zw}(s)$, has the following property:

$$\|T_{zw}\|_2 = \{\operatorname{trace}(E^{\mathrm{T}}PE)$$
$$+ \operatorname{trace}\left[\left(A^{\mathrm{T}}P + PA + C_2^{\mathrm{T}}C_2\right)Q\right]\}^{\frac{1}{2}}$$
$$(16)$$

Similarly, consider the standard LQG problem for the following system:

$$\dot{x} = Ax + Bu + G_\star d \qquad (17)$$

$$y = Cx + N_\star n, \quad N_\star > 0 \qquad (18)$$

$$z = \begin{pmatrix} H_\star x \\ R_\star u \end{pmatrix}, \quad R_\star > 0, \quad w = \begin{pmatrix} d \\ n \end{pmatrix} \qquad (19)$$

where x is the state, u is the control, d and n white noises with identity covariance, and y the measurement output. It is assumed that (A, B) is stabilizable and (A, C) is detectable. The control objective is to design an appropriate control law that minimizes the expectation of $|z|^2$. Such an LQG problem can be solved via the H_2 optimal control problem for the following auxiliary system Σ_{LQG} (see Doyle 1983):

$$\dot{x} = Ax + Bu + [\, G_\star \quad 0 \,]w \qquad (20)$$

$$y = Cx + [\, 0 \quad N_\star]w \qquad (21)$$

$$z = \begin{pmatrix} H_\star \\ 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ R_\star \end{pmatrix} u \qquad (22)$$

H_2 optimal control problem for discrete-time systems can be solved in a similar way via the corresponding discrete-time algebraic Riccati equations. It is worth noting that many works can be found in the literature that deal with solutions to discrete-time algebraic Riccati equations related to optimal control problems; see, for example, Kucera (1972), Pappas et al. (1980), and Silverman (1976), to name a few. It is proven in Chen et al. (1994b) that solutions to the discrete- and continuous-time algebraic Riccati equations for optimal control problems can be unified. More specifically, the solution to a discrete-time Riccati equation can be done through solving an equivalent continuous-time one and vice versa.

Singular Case

As in the previous section, only the key procedure in solving the singular H_2-optimization problem for continuous-time systems is addressed. For the singular problem, it is generally not possible to obtain an optimal solution, except for some situations when the given plant satisfies certain geometric constraints; see, e.g., Chen et al. (1993) and Stoorvogel et al. (1993). It is more feasible

to find a suboptimal control law for the singular problem, i.e., to find an appropriate control law such that the H_2 norm of the resulting closed-loop transfer matrix from w to z can be made arbitrarily close to the best possible performance. The procedure given below is to transform the original problem into an H_2 almost disturbance decoupling problem; see Stoorvogel (1992) and Stoorvogel et al. (1993).

Consider the given plant in Eqs. (1)–(3) with Assumption 1 and/or Assumption 2 not satisfied. First, find the largest solution $P \geq 0$ for the following linear matrix inequality

$$F(P) = \begin{pmatrix} A^{\mathsf{T}}P + PA + C_2^{\mathsf{T}}C_2 & PB + C_2^{\mathsf{T}}D_2 \\ B^{\mathsf{T}}P + D_2^{\mathsf{T}}C_2 & D_2^{\mathsf{T}}D_2 \end{pmatrix} \geq 0 \tag{23}$$

and find the largest solution $Q \geq 0$ for

$$G(Q) = \begin{pmatrix} AQ + QA^{\mathsf{T}} + EE^{\mathsf{T}} & QC_1^{\mathsf{T}} + ED_1^{\mathsf{T}} \\ C_1 Q + D_1 E^{\mathsf{T}} & D_1 D_1^{\mathsf{T}} \end{pmatrix} \geq 0 \tag{24}$$

Note that by decomposing the quadruples (A, B, C_2, D_2) and (A, E, C_1, D_1) into various subsystems in accordance with their structural properties, solutions to the above linear matrix inequalities can be obtained by solving a Riccati equation similar to those in Eq. (5) or Eq. (5) for the regular case. In fact, for the regular problem, the largest solution $P \geq 0$ for Eq. (23) and the stabilizing solution $P \geq 0$ for Eq. (5) are identical. Similarly, the largest solution $Q \geq 0$ for Eq. (24) and the stabilizing solution $Q \geq 0$ for Eq. (13) are also the same. Interested readers are referred to Stoorvogel et al. (1993) for more details or to Chen et al. (2004) for a more systematic treatment on the structural decomposition of linear systems and its connection to the solutions of the linear matrix inequalities.

It can be shown that the best achievable H_2 norm of the closed-loop transfer matrix from w to z, i.e., the best possible performance over all internally stabilizing control laws, is given by

$$\gamma_2^\star = \{ \mathrm{trace}(E^{\mathsf{T}}PE)$$
$$+ \mathrm{trace}\left[\left(A^{\mathsf{T}}P + PA + C_2^{\mathsf{T}}C_2 \right) Q \right] \}^{\frac{1}{2}} \tag{25}$$

Next, partition

$$F(P) = \begin{pmatrix} C_{\mathrm{P}}^{\mathrm{T}} \\ D_{\mathrm{P}}^{\mathrm{T}} \end{pmatrix} (C_{\mathrm{P}} \quad D_{\mathrm{P}})$$

$$\text{and } G(Q) = \begin{pmatrix} E_{\mathrm{Q}} \\ D_{\mathrm{Q}} \end{pmatrix} (E_{\mathrm{Q}}^{\mathrm{T}} \quad D_{\mathrm{Q}}^{\mathrm{T}}) \qquad (26)$$

where $[C_{\mathrm{P}} \quad D_{\mathrm{P}}]$ and $[E_{\mathrm{Q}}^{\mathrm{T}} \quad D_{\mathrm{Q}}^{\mathrm{T}}]$ are of maximal rank, and then define an auxiliary system Σ_{PQ}:

$$\dot{x}_{\mathrm{PQ}} = A x_{\mathrm{PQ}} + B u + E_{\mathrm{Q}} w_{\mathrm{PQ}} \qquad (27)$$

$$y = C_1 x_{\mathrm{PQ}} + D_{\mathrm{Q}} w_{\mathrm{PQ}} \qquad (28)$$

$$z_{\mathrm{PQ}} = C_{\mathrm{P}} x_{\mathrm{PQ}} + D_{\mathrm{P}} u \qquad (29)$$

It can be shown that the quadruple $(A, B, C_{\mathrm{P}}, D_{\mathrm{P}})$ is right invertible and has no invariant zeros in the open right-half complex plane, and the quadruple $(A, E_{\mathrm{Q}}, C_1, D_{\mathrm{Q}})$ is left invertible and has no invariant zeros in the open right-half complex plane. It can also be shown that there exists an appropriate control law such that when it is applied to Σ_{PQ}, the resulting closed-loop system is internally stable and the H_2 norm of the closed-loop transfer matrix from w_{PQ} to z_{PQ} can be made arbitrarily small. Equivalently, H_2 almost disturbance decoupling problem for Σ_{PQ} is solvable.

More importantly, it can further be shown that if an appropriate control law solves the H_2 almost disturbance decoupling problem for Σ_{PQ}, then it solves the H_2 suboptimal problem for Σ. As such, the solution to the singular H_2 control problem for Σ can be done by finding a solution to the H_2 almost disturbance decoupling problem for Σ_{PQ}. There are vast results available in the literature dealing with disturbance decoupling problems. More detailed treatments can be found in Saberi et al. (1995).

Conclusion

This entry considers the basic solutions to H_2 optimal control problems for continuous-time systems. Both the regular problem and the general singular problem are presented. Readers interested in more details are referred to Saberi et al. (1995) and the references therein, for the complete treatment of H_2 optimal control problems, and to Chap. 10 of Chen et al. (2004) for the unification and differentiation of H_2 control, H_∞ control, and disturbance decoupling control problems. H_2 optimal control is a mature area and has a long history. Possible future research includes issues on how to effectively utilize the theory in solving real-life problems.

Cross-References

▶ H-Infinity Control
▶ Linear Matrix Inequality Techniques in Optimal Control
▶ Linear Quadratic Optimal Control
▶ Optimal Control via Factorization and Model Matching
▶ Stochastic Linear-Quadratic Control

Bibliography

Anderson BDO, Moore JB (1989) Optimal control: linear quadratic methods. Prentice Hall, Englewood Cliffs

Bryson AE, Ho YC (1975) Applied optimal control, optimization, estimation, and control. Wiley, New York

Chen BM, Saberi A (1993) Necessary and sufficient conditions under which an H_2-optimal control problem has a unique solution. Int J Control 58:337–348

Chen BM, Saberi A, Sannuti P, Shamash Y (1993) Construction and parameterization of all static and dynamic H_2-optimal state feedback solutions, optimal fixed modes and fixed decoupling zeros. IEEE Trans Autom Control 38:248–261

Chen BM, Saberi A, Shamash Y, Sannuti P (1994a) Construction and parameterization of all static and dynamic H_2-optimal state feedback solutions for discrete time systems. Automatica 30:1617–1624

Chen BM, Saberi A, Shamash Y (1994b) A non-recursive method for solving the general discrete time algebraic Riccati equation related to the H_∞ control problem. Int J Robust Nonlinear Control 4:503–519

Chen BM, Saberi A, Shamash Y (1996) Necessary and sufficient conditions under which a discrete time H_2-optimal control problem has a unique solution. J Control Theory Appl 13:745–753

Chen BM, Lin Z, Shamash Y (2004) Linear systems theory: a structural decomposition approach. Birkhäuser, Boston

Doyle JC (1983) Synthesis of robust controller and filters. In: Proceedings of the 22nd IEEE conference on decision and control, San Antonio

Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York

Geerts T (1989) All optimal controls for the singular linear quadratic problem without stability: a new interpretation of the optimal cost. Linear Algebra Appl 122:65–104

Kailath T (1974) A view of three decades of linear filtering theory. IEEE Trans Inf Theory 20: 146–180

Kucera V (1972) The discrete Riccati equation of optimal control. Kybernetika 8:430–447

Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York

Lewis FL (1986) Optimal control. Wiley, New York

Pappas T, Laub AJ, Sandell NR Jr (1980) On the numerical solution of the discrete-time algebraic Riccati equation. IEEE Trans Autom Control AC-25:631–641

Saberi A, Sannuti P, Chen BM (1995) H_2 optimal control. Prentice Hall, London

Silverman L (1976) Discrete Riccati equations: alternative algorithms, asymptotic properties, and system theory interpretations. Control Dyn Syst 12:313–386

Stoorvogel AA (1992) The singular H_2 control problem. Automatica 28:627–631

Stoorvogel AA, Saberi A, Chen BM (1993) Full and reduced order observer based controller design for H_2-optimization. Int J Control 58:803–834

Trentelman HL, Stoorvogel AA (1995) Sampled-data and discrete-time H_2 optimal control. SIAM J Control Optim 33:834–862

Willems JC, Kitapci A, Silverman LM (1986) Singular optimal control: a geometric approach. SIAM J Control Optim 24:323–337

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

H-Infinity Control

Keith Glover
Department of Engineering, University of Cambridge, Cambridge, UK

Abstract

The area of robust control, where the performance of a feedback system is designed to be robust to uncertainty in the plant being controlled, has received much attention since the 1980s. System analysis and controller synthesis based on the H-infinity norm has been central to progress in this area. This article outlines how the control law that minimizes the H-infinity norm of the closed-loop system can be derived. Connections to other problems, such as game theory and risk-sensitive control, are discussed and finally appropriate problem formulations to produce "good" controllers using this methodology are outlined.

Keywords

Loop-shaping; Robust control; Robust stability

Introduction

The $\mathcal{H}_\infty$-norm probably first entered the study of robust control with the observations made by Zames (1981) in the considering optimal sensitivity. The so-called $\mathcal{H}_\infty$ methods were subsequently developed and are now routinely available to control engineers. In this entry we consider the $\mathcal{H}_\infty$ methods for control, and for simplicity of exposition, we will restrict our attention to linear, time-invariant, finite dimensional, continuous-time systems. Such systems can be represented by their transfer function matrix, $G(s)$, which will then be a rational function of s. Although the Hardy Space, $\mathcal{H}_\infty$, also includes nonrational functions, a rational $G(s)$ is in $\mathcal{H}_\infty$ if and only if it is proper and all its poles are in the open left half plane, in which case the $\mathcal{H}_\infty$-norm is defined as:

$$\|G(s)\|_\infty = \sup_{\mathrm{Re}\,s>0} \sigma_{\max}(G(s)) = \sup_{-\infty<\omega<\infty} \sigma_{\max}((j\omega))$$

(where $\sigma_{\max}$ denotes the largest singular value). Hence for a single input/single output system with transfer function, $g(s)$, its $\mathcal{H}_\infty$-norm, $\|g(s)\|_\infty$ gives the maximum value of $|g(j\omega)|$ and hence the maximum amplification of sinusoidal signals by a system with this transfer function. In the multi-input/multi-output case a similar result holds regarding the system amplification of a vector of sinusoids. There is now a good collection of graduate level textbooks that cover the area in some detail from a variety of approaches, and these are listed

in the Recommended Reading section and the references in this article are generally to these texts rather than to the original journal papers.

Consider a system with transfer function, $G(s)$, input vector, $u(t) \in \mathcal{L}_2(0,\infty)$ and an output vector, $y(t)$, whose Laplace transforms are given by $\bar{u}(s)$ and $\bar{y}(s)$. Such a system will have a state space realization,

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

giving $G(s) = D + C(sI - A)^{-1}B$, which we also denote

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right],$$

and hence $\bar{y}(s) = G(s)\bar{u}(s)$ if $x(0) = 0$.

There are two main reasons for using the $\mathcal{H}_\infty$-norm. Firstly in representing the system gain for input signals $u(t) \in \mathcal{L}_2(0,\infty)$ or equivalently $\bar{u}(j\omega) \in \mathcal{L}_2(-\infty,\infty)$, with corresponding norm $\|u\|_2^2 = \int_0^\infty u(t)^*u(t)\,dt$ (where x^* denotes the conjugate transpose of the vector x (or a matrix)). With these input and output spaces the induced norm of the system is easily shown to be the $\mathcal{H}_\infty$-norm of $G(s)$, and in particular,

$$\|y\|_2 \leq \|G(s)\|_\infty \|u\|_2$$

Hence in a control context the $\mathcal{H}_\infty$-norm can give a measure of the gain, for example, from disturbances to the resulting errors. In the interconnection of systems, the property that $\|P(s)Q(s)\|_\infty \leq \|P(s))\|_\infty \|Q(s)\|_\infty$ is often useful.

The second reason for using the $\mathcal{H}_\infty$-norm is in representing uncertainty in the plant being controlled, e.g., the nominal plant is $P_o(s)$ but the actual plant is $P(s) = P_o(s) + \Delta(s)$ where $\|\Delta(s)\|_\infty \leq \delta$.

A typical control design problem is given in Fig. 1, i.e.,

$$\begin{bmatrix} \bar{z} \\ \bar{y} \end{bmatrix} = P\begin{bmatrix} \bar{w} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} P_{11}\bar{w} + P_{12}\bar{u} \\ P_{21}\bar{w} + P_{22}\bar{u} \end{bmatrix}$$
$$\bar{u} = K\bar{y}$$
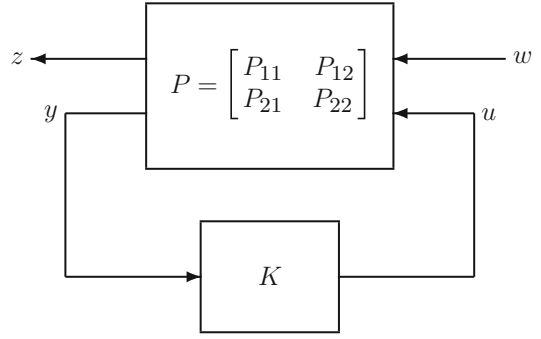
H-Infinity Control, Fig. 1 Lower linear fractional transformation: feedback system

$$\Rightarrow \bar{y} = (I - P_{22}K)^{-1}P_{21}\bar{w},$$
$$\bar{u} = K(I - P_{22}K)^{-1}P_{21}\bar{w}$$
$$\bar{z} = \left(P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}\right)\bar{w}$$
$$=: \mathcal{F}_l(P,K)\bar{w} =: T_{z\leftarrow w}\bar{w}$$

where $\mathcal{F}_l(P,K)$ denotes the lower Linear Fractional Transformation (LFT) with connection around the lower terminals of P as in Fig. 1.

The standard $\mathcal{H}_\infty$-control synthesis problem is to find a controller with transfer function, K, that

> stabilizes the closed-loop system in Fig. 1 and minimizes $\|\mathcal{F}_l(P,K)\|_\infty$.

That is, the controller is designed to minimize the worst-case effect of the disturbance w on the output/error signal z as measured by the $\mathcal{L}_2$ norm of the signals. This article will describe the solution to this problem.
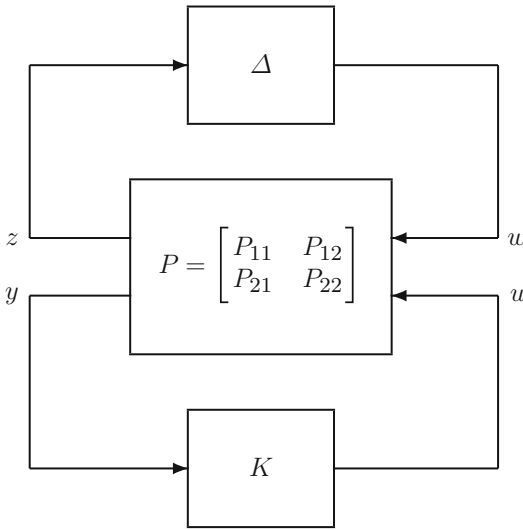
Robust Stability

Before we describe the solution to the synthesis problem, consider the problem of the robust stability of an uncertain plant with a feedback controller. Suppose the plant is given by the upper LFT, $\mathcal{F}_u(P,\Delta)$ with $\|\Delta\|_\infty \leq 1/\gamma$ as illustrated in Fig. 2,

$$\bar{y} = \mathcal{F}_u(P,\Delta)\bar{u}, \tag{1}$$
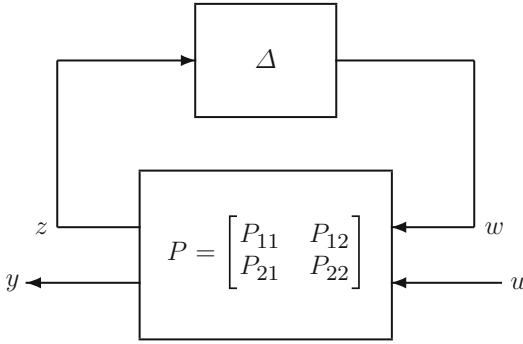
H-Infinity Control, Fig. 2 Upper linear fractional transformation

H-Infinity Control, Fig. 3 Feedback system with plant uncertainty

where $\mathcal{F}_u(P,K) := P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12}$

$$\text{(2)}$$

The *small gain theorem* then states that the feedback system of Fig. 3 will be stable for all such Δ if the feedback connection of P_{22} and K is stable and $\|\mathcal{F}_l(P,K)\|_\infty < \gamma$. This robust stability result is valid if P and Δ are both stable; more care is required when either or both are unstable but with such care a similar result is true.

Let us consider a couple of examples. First suppose that the uncertainty is represented as output multiplicative uncertainty,

$$P_\Delta = (I + W_1 \Delta W_2) P_o = \mathcal{F}_u\left(\begin{bmatrix} 0 & W_2 P_o \\ W_1 & P_o \end{bmatrix}, \Delta\right)$$

with robust stability test given by

$$\|\mathcal{F}_l\left(\begin{bmatrix} 0 & W_2 P_o \\ W_1 & P_o \end{bmatrix}, K\right)\|_\infty$$
$$= \|W_2 P_o K (I - P_o K)^{-1} W_1\|_\infty < \gamma$$

As a second example consider the plants $P_\Delta = (\tilde{M} + \Delta_M)^{-1}(\tilde{N} + \Delta_N)$, with $\Delta = \begin{bmatrix} \Delta_N & \Delta_M \end{bmatrix}$ and $\|\Delta\|_\infty \leq 1/\gamma$. Here $P_o = \tilde{M}^{-1}\tilde{N}$ is a left coprime factorization of the nominal plant and the plants P_Δ are represented by perturbations to these coprime factors. In this case $P_\Delta = \mathcal{F}_u(P, \Delta)$, where

$$P = \begin{bmatrix} \begin{bmatrix} 0 \\ -\tilde{M}^{-1} \\ \tilde{M}^{-1} \end{bmatrix} & \begin{bmatrix} I \\ -\tilde{M}^{-1}\tilde{N} \\ \tilde{M}^{-1}\tilde{N} \end{bmatrix} \end{bmatrix}$$

and the robust stability test will be

$$\|\mathcal{F}_l(P,K)\|_\infty = \left\|\begin{bmatrix} K \\ -I \end{bmatrix}(I - P_o K)^{-1}\tilde{M}^{-1}\right\|_\infty < \gamma$$

This is related to plant perturbations in the gap metric (see Vinnicombe 2001). It is therefore observed that the robust stability test for these useful representations of uncertain plants is given by an $\mathcal{H}_\infty$-norm test just as in the controller synthesis problem.

Derivation of the $\mathcal{H}_\infty$-Control Law

In this section we present a solution to the $\mathcal{H}_\infty$-control problem and give some interpretations of the solution. The approach presented is as in by Doyle et al. (1989); see also Zhou et al. (1996). We will make some simplifying structural assumptions to make the formulae less complex and will *not* state the required assumptions on rank, stabilizability, and detectability. Let the system in Fig. 1 be described by the equations:

$$\dot{x}(t) = Ax(t) + B_1 w(t) + B_2 u(t) \qquad \text{(3)}$$

$$z(t) = C_1 x(t) + D_{12} u(t) \qquad (4)$$

$$y(t) = C_2 x(t) + D_{21} w(t) \qquad (5)$$

i.e., in Fig. 1

$$P = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array} \right]$$

where we also assume, with little loss of generality, that $D_{12}^* D_{12} = I$, $D_{21} D_{21}^* = I$, $D_{12}^* C_1 = 0$ and $B_1 D_{21}^* = 0$. Since we wish to have $\|T_{z \leftarrow w}\|_\infty < \gamma$, we need to find u such that

$$\|z\|_2^2 - \gamma^2 \|w\|_2^2 < 0 \text{ for all } w \neq 0 \in \mathcal{L}_2(0, \infty).$$

We could consider w to be an adversary trying to make this expression positive, while u has to ensure that it always remains negative in spite of the malicious intentions of w, as in a noncooperative game. Suppose that there exists a solution, X_∞, to the Algebraic Riccati Equation (ARE),

$$A^* X_\infty + X_\infty A + C_1^* C_1$$
$$+ X_\infty (\gamma^{-2} B_1 B_1^* - B_2 B_2^*) X_\infty = 0 \quad (6)$$

with $X_\infty \geq 0$ and $A + (\gamma^{-2} B_1 B_1^* - B_2 B_2^*) X_\infty$ a stable "A-matrix." A simple substitution then gives that

$$\frac{d}{dt}(x(t)^* X_\infty x(t)) = -z^* z + \gamma^2 w^* w$$
$$+ v^* v - \gamma^2 r^* r$$

where

$$v := u + B_2^* X_\infty x, \quad r := w - \gamma^{-2} B_1^* X_\infty x.$$

Now let $x(0) = 0$ and assuming stability so that $x(\infty) = 0$, then integrating from 0 to ∞ gives

$$\|z\|_2^2 - \gamma^2 \|w\|_2^2 = \|v\|_2^2 - \gamma^2 \|r\|_2^2 \qquad (7)$$

If the state is available to u, then the control law $u = -B_2^* X_\infty x$ gives $v = 0$ and $\|z\|_2^2 - \gamma^2 \|w\|_2^2 < 0$ for all $w \neq 0$. It can be shown that (6) has a solution if there exists a controller

such that $\|\mathcal{F}_l(P, K)\|_\infty < \gamma$. In addition since transposing a system does not change its $\mathcal{H}_\infty$-norm, the following dual ARE will also have a solution, $Y_\infty \geq 0$,

$$AY_\infty + Y_\infty A^* + B_1 B_1^*$$
$$+ Y_\infty(\gamma^{-2} C_1^* C_1 - C_2^* C_2) Y_\infty = 0 \quad (8)$$

To obtain a solution to the output feedback case, note that (7) implies that $\|z\|_2^2 < \gamma^2 \|w\|_2^2$ if and only if $\|v\|_2^2 < \gamma^2 \|r\|_2^2$ and $\bar{v} = \mathcal{F}_l(P_{\text{tmp}}, K)\bar{r}$ where

$$\begin{bmatrix} \bar{v} \\ \bar{y} \end{bmatrix} = P_{\text{tmp}} \begin{bmatrix} \bar{r} \\ \bar{u} \end{bmatrix},$$

and

$$P_{\text{tmp}} = \left[\begin{array}{c|cc} A + \gamma^{-2} B_1 B_1^* X_\infty & B_1 & B_2 \\ \hline B_2^* X_\infty & 0 & I \\ C_2 & D_{21} & 0 \end{array} \right]$$

The special structure of this problem enables a solution to be derived in much the same way as the dual of the state feedback problem. The corresponding ARE will have a solution $Y_{\text{tmp}} = (I - \gamma^{-2} Y_\infty X_\infty)^{-1} Y_\infty \geq 0$ if and only if the spectral radius $\rho(Y_\infty X_\infty) < \gamma^2$.

The above outline, supported by significant technical detail and assumptions, will therefore demonstrate that there exists a stabilizing controller, $K(s)$, such that the system described by (3–1) satisfies $\|T_{z \leftarrow w}\|_\infty < \gamma$ if and only if there exist stabilizing solutions to the AREs in (6) and (8) such that

$$X_\infty \geq 0, \ \ Y_\infty \geq 0, \ \ \rho(Y_\infty X_\infty) < \gamma^2 \qquad (9)$$

The state equations for the resulting controller can be written as

$$\dot{\hat{x}} = A\hat{x} + B_1 \hat{w}_{\text{worst}} + B_2 u + Z_\infty L_\infty (C_2 \hat{x} - y)$$

$$u = F_\infty \hat{x}, \ \ \hat{w}_{\text{worst}} = \gamma^{-2} B_1^* X_\infty \hat{x}$$

$$F_\infty := -B_2^* X_\infty, \ \ L_\infty := -Y_\infty C_2^*,$$

$$Z_\infty := (I - \gamma^{-2} Y_\infty X_\infty)^{-1}$$

giving feedback from a state estimator in the presence of an estimate of the worst-case disturbance.

As $\gamma \to \infty$ the standard LQG controller is obtained with state feedback of a state estimate obtained from a Kalman filter. In contrast to the LQG problem, the controller depends on the value of γ, and if this is chosen to be too small, then one of the conditions in (9) will be violated. In order to determine the minimum achievable value of γ, a bisection search over γ can be performed checking (9) for each candidate value of γ.

In the limit as $\gamma \to \gamma_{\text{opt}}$ (its minimum value), a variety of situations can arise and the formulae given here may become ill-conditioned. Typically achieving γ_{opt} is more of an interesting and sometimes challenging mathematical exercise rather than a control system requirement.

This control problem does not have a unique solution, and all solutions can be characterized by an LFT form such as $K = \mathcal{F}_l(M, Q)$ where $Q \in \mathcal{H}_\infty$ with $\|Q\|_\infty < 1$, the present solution is sometimes referred to as the "central solution" obtained with $Q = 0$.

Relations for Other Solution Methods and Problem Formulations

The $\mathcal{H}_\infty$-control problem has been shown to be related to an extraordinarily wide variety of mathematical techniques and to other problem areas, and investigations of these connections have been most fruitful. Earlier approaches (see Francis 1988) firstly used the characterization of all stabilizing controllers of Youla et al. (see Vidyasagar 1985) which shows that all stable closed-loop systems can be written as

$$\mathcal{F}_l(P, K) = T_1 + T_2 Q T_3, \text{ where } Q \in \mathcal{H}_\infty$$

and then solved the model matching problem $\inf_{Q \in \mathcal{H}_\infty} \|T_1 + T_2 Q T_3\|_\infty$. This model matching problem is related to interpolation theory and resulted in a productive interaction with the operator theory. One solution method reduces this problem to J-spectral factorisation problems

$\left(\text{where } J = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}\right)$ and generates state-space solutions to these problems (Kimura 1997).

The derivation above clearly demonstrates relations to noncooperative differential games, and this is fully developed in Başar and Bernhard (1995) and Green and Limebeer (1995).

The model matching problem is clearly a convex optimization problem. The solution of linear matrix inequalities can give effective methods for solving certain convex optimization problems (e.g., calculating the $\mathcal{H}_\infty$ norm using the bounded real lemma) and can be exploited in the $\mathcal{H}_\infty$-control problem. See Boyd and Barratt (1991) for a variety of results on convex optimization and control and Dullerud and Paganini (2000) for this approach in robust control.

As noted above there is a family of solutions to the $\mathcal{H}_\infty$-control problem. The central solution in fact minimizes the entropy integral given by

$$I(T_{z \leftarrow w}; \gamma) := -\frac{\gamma^2}{2\pi} \int_{-\infty}^{\infty} \ln$$
$$\left| \det(I - \gamma^{-2} T_{z \leftarrow w}(j\omega)^* T_{z \leftarrow w}(j\omega)) \right| \, d\omega \tag{10}$$

It can be seen that this criterion will penalize the singular values of $T_{z \leftarrow w}(j\omega)$ from being close to γ for a large range of frequencies.

One of the more surprising connections is with the risk-sensitive stochastic control problem (Whittle 1990) where w is assumed to be Gaussian white noise and it is desired to minimize

$$J_T(\gamma) := \frac{\gamma^2}{T} \ln \mathbf{E} \left\{ e^{\frac{1}{2}\gamma^{-2} V_T} \right\} \tag{11}$$

$$\text{where } V_T := \int_{-T}^{T} z(t)^* z(t) \, dt \tag{12}$$

The situation with $\gamma^2 > 0$ corresponds to the risk averse controller since large values of V_T are heavily penalized by the exponential function. It can be shown that if $\|T_{z \leftarrow w}\|_\infty < \gamma$, then

$$\lim_{T \to \infty} J_T(\gamma) = I(T_{z \leftarrow w}; \gamma)$$

and hence the central controller minimizes both the entropy integral and the risk-sensitive cost

function. When γ is chosen to be too small, Whittle refers to the controller having a "neurotic breakdown" because the cost will be infinite for all possible control laws! If in (11) we set $\gamma^2 = -\theta^{-1}$, then the entropy minimizing controller will have $\theta < 0$ and will be risk-averse. The risk neutral controller is when $\theta \to 0$, $\gamma \to \infty$ and gives the standard LQG case. If $\theta > 0$, then the controller will be risk-seeking, believing that large variance will be in its favor.

Controller Design with $\mathcal{H}_\infty$ Optimization

The above solutions to the $\mathcal{H}_\infty$ mathematical problem do not give guidance on how to set up a problem to give a "good" control system design. The problem formulation typically involves identifying frequency-dependent weighting matrices to characterize the disturbances, w, and the relative importance of the errors, z (see Skogestad and Postlethwaite 1996). The choice of weights should also incorporate system uncertainty to obtain a robust controller.

One approach that combines both closed-loop system gain and system uncertainty is called $\mathcal{H}_\infty$ *loop-shaping* where the desired closed-loop behavior is determined by the design of the loop-shape using pre- and post-compensators and the system uncertainty is represented in the gap metric (see Vinnicombe 2001). This makes classical criteria such as low frequency tracking error, bandwidth, and high-frequency roll-off all easily incorporated. In this framework the performance and robustness measures are very well matched to each other. Such an approach has been successfully exploited in a number of practical examples (e.g., Hyde (1995) for flight control taken through to successful flight tests). Standard control design software packages now routinely have $\mathcal{H}_\infty$-control design modules.

Summary and Future Directions

We have outlined the derivation of $\mathcal{H}_\infty$ controllers with straightforward assumptions that

nevertheless exhibit most of the features of linear time-invariant systems without such assumptions and for which routine design software is now available. Connections to a surprisingly large range of other problems are also discussed.

Generalizations to more general cases such as time-varying and nonlinear systems, where the norm is interpreted as the induced norm of the system in $\mathcal{L}_2$, can be derived although the computational aspects are no longer routine. For the problems of robust control, there are necessarily continuing efforts to match the mathematical representation of system uncertainty and system performance to the physical system requirements and to have such representations amenable to analysis and computation.

Cross-References

▶ Fundamental Limitation of Feedback Control
▶ H₂ Optimal Control
▶ Linear Quadratic Optimal Control
▶ LMI Approach to Robust Control
▶ Robust $\mathcal{H}_2$ Performance in Feedback Control
▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Bibliography

Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York

Green M, Limebeer D (1995) Linear robust control. Prentice Hall, Englewood Cliffs

Başar T, Bernhard P (1995) H^∞-optimal control and related minimax design problems, 2nd edn. Birkhäuser, Boston

Boyd SP, Barratt CH (1991) Linear controller design: limits of performance. Prentice Hall, Englewood Cliffs

Doyle JC, Glover K, Khargonekar PP, Francis BA (1989) State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. IEEE Trans Autom Control 34(8):831–847

Francis BA (1988) A course in $\mathcal{H}_\infty$ control theory. Lecture notes in control and information sciences, vol 88. Springer, Berlin, Heidelberg

Hyde RA (1995) $\mathcal{H}_\infty$ aerospace control design: a VSTOL flight application. Springer, London

Kimura H (1997) Chain-scattering approach to $\mathcal{H}_\infty$-control. Birkhäuser, Basel

Skogestad S, Postlethwaite I (1996) Multivariable feedback control: analysis and design. Wiley, Chichester

Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge

Vinnicombe G (2001) Uncertainty and feedback: $\mathcal{H}_\infty$ loop-shaping and the ν-gap metric. Imperial College Press, London

Whittle P (1990) Risk-sensitive optimal control. Wiley, Chichester

Zames G (1981) Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses. IEEE Trans Automat Control 26:301–320

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River, New Jersey

History of Adaptive Control

Karl Åström
Department of Automatic Control, Lund
University, Lund, Sweden

Abstract

This entry gives an overview of the development of adaptive control, starting with the early efforts in flight and process control. Two popular schemes, the model reference adaptive controller and the self-tuning regulator, are described with a thumbnail overview of theory and applications. There is currently a resurgence in adaptive flight control as well as in other applications. Some reflections on future development are also given.

Keywords

Adaptive control; Auto-tuning; Flight control; History; Model reference adaptive control; Process control; Robustness; Self-tuning regulators; Stability

Introduction

In everyday language, *to adapt* means to change a behavior to conform to new circumstances, for example, when the pupil area changes to accommodate variations in ambient light. The distinction between adaptation and conventional feedback is subtle because feedback also attempts to reduce the effects of disturbances and plant uncertainty. Typical examples are *adaptive optics* and *adaptive machine tool control* which are conventional feedback systems, with controllers having constant parameters. In this entry we take the pragmatic attitude that an adaptive controller is a controller that can modify its *behavior* in response to changes in the dynamics of the process and the character of the disturbances, by adjusting the controller parameters.

Adaptive control has had a colorful history with many ups and downs and intense debates in the research community. It emerged in the 1950s stimulated by attempts to design autopilots for supersonic aircrafts. Autopilots based on constant-gain, linear feedback worked well in one operating condition but not over the whole flight envelope. In process control there was also a need for automatic tuning of simple controllers.

Much research in the 1950s and early 1960s contributed to conceptual understanding of adaptive control. Bellman showed that *dynamic programming* could capture many aspects of adaptation (Bellman 1961). Feldbaum introduced the notion of *dual control*, meaning that control should be probing as well as directing; the controller should thus inject test signals to obtain better information. Tsypkin showed that schemes for *learning and adaptation* could be captured in a common framework (Tsypkin 1971).

Gabor's work on adaptive filtering (Gabor et al. 1959) inspired Widrow to develop an analogue neural network (Adaline) for adaptive control (Widrow 1962). Widrow's adaptation mechanism was inspired by Hebbian learning in biological systems (Hebb 1949).

There are adaptive control problems in economics and operations research. In these fields the problems are often called *decision making under uncertainty*. A simple idea, called the *certainty equivalence principle* proposed by Simon (1956), is to neglect uncertainty and treat estimates as if they are true. Certainty equivalence was commonly used in early work on adaptive control.

A period of intense research and ample funding ended dramatically in 1967 with a crash of the rocket powered X15-3 using Honeywell's MH-96 self-oscillating adaptive controller. The self-oscillating adaptive control system has, however, been successfully used in several missiles.

Research in adaptive control resurged in the 1970s, when the two schemes the model reference adaptive control (MRAC) and the self-tuning regulator (STR) emerged together with successful applications. The research was influenced by stability theory and advances in the field of system identification. There was an intensive period of research from the late 1970s through the 1990s. The insight and understanding of stability, convergence, and robustness increased. Recently there has been renewed interest because of flight control (Hovakimyan and Cao 2010; Lavretsky and Wise 2013) and other applications; there is, for example a need for adaptation in autonomous systems.

The Brave Era

Supersonic flight posed new challenges for flight control. Eager to obtain results, there was a very short path from idea to flight test with very little theoretical analysis in between. A number of research projects were sponsored by the US air force. Adaptive flight control systems were developed by General Electric, Honeywell, MIT, and other groups. The systems are documented in the Self-Adaptive Flight Control Systems Symposium held at the Wright Air Development Center in 1959 (Gregory 1959) and the book (Mishkin and Braun 1961).

Whitaker of the MIT team proposed the model reference adaptive controller system which is based on the idea of specifying the performance of a servo system by a reference. Honeywell proposed a self-oscillating adaptive system (SOAS) which attempted to keep a given gain margin by bringing the system to self-oscillation. The system was flight-tested on several aircrafts. It experienced a disaster in a test on the X-15. Combined with the success of gain scheduling

based on air data sensors, the interest in adaptive flight control diminished significantly.

There was also interest of adaptation for process control. Foxboro patented an adaptive process controller with a pneumatic adaptation mechanism in 1950 (Foxboro 1950). DuPont had joint studies with IBM aimed at computerized process control. Kalman worked for a short time at the Engineering Research Laboratory at DuPont, where he started work that led to a paper (Kalman 1958), which is the inspiration of the self-tuning regulator. The abstract of this entry has the statement, *This paper examines the problem of building a machine which adjusts itself automatically to control an arbitrary dynamic process*, which clearly captures the dream of early adaptive control.

Draper and Li investigated the problem of operating aircraft engines optimally, and they developed a self-optimizing controller that would drive the system towards optimal working conditions. The system was successfully flight-tested (Draper and Li 1966) and initiated the field of *extremal control*.

Many of the ideas that emerged in the brave era inspired future research in adaptive control. The MRAC, the STR, and extremal control are typical examples.

Model Reference Adaptive Control (MRAC)

The MRAC was one idea from the early work on flight control that had a significant impact on adaptive control. A block diagram of a system with model reference adaptive control is shown in Fig. 1. The system has an ordinary feedback loop with a controller, having adjustable parameters, and the process. There is also a reference model which gives the ideal response y_m to the command signal y_m and a mechanism for adjusting the controller parameters θ. The parameter adjustment is based on the process output y, the control signal u, and the output y_m of the reference model. Whitaker proposed the following rule for adjusting the parameters:
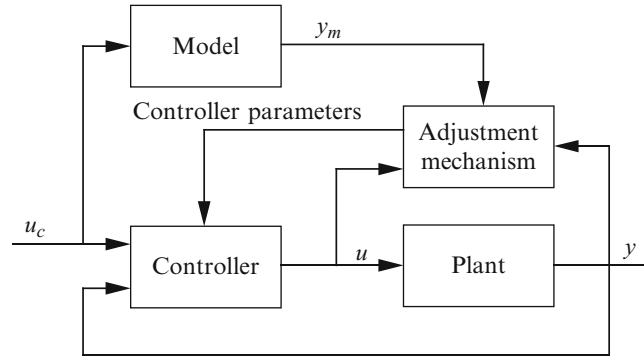
$$\frac{d\theta}{dt} = -\gamma e\,\frac{\partial e}{\partial \theta}, \tag{1}$$

where $e = y - y_m$ and $\partial e / \partial \theta$ is the sensitivity derivative. Efficient ways to compute the sensitivity derivative were already available in sensitivity theory. The adaptation law (1) became known as the *MIT rule*.

Experiments and simulations of the model reference adaptive systems indicated that there could be problems with instability, in particular if the adaptation gain γ in Eq. (1) is large. This observation inspired much theoretical research. The goal was to replace the MIT rule by other parameter adjustment rules with guaranteed stability; the models used were non linear continuous time differential equations. The papers Butchart and Shackcloth (1965) and Parks (1966) demonstrated that control laws could be obtained using Lyapunov theory. When all state variables are measured, the adaptation laws obtained were similar to the MIT rule (1), but the sensitivity function was replaced by linear combinations of states and control variables. The problem was more difficult for systems that only permitted output feedback. Lyapunov theory could still be used if the process transfer function was strictly positive real, establishing a connection with Popov's hyper-stability theory (Landau 1979). The assumption of a positive real process is a severe restriction because such systems can be successfully controlled by high-gain feedback. The difficulty was finally resolved by using a scheme called *error augmentation* (Monopoli 1974; Morse 1980).

There was much research, and by the late 1980s, there was a relatively complete theory for MRAC and a large body of literature (Anderson et al. 1986; Åström and Wittenmark 1989; Egardt 1979; Goodwin and Sin 1984; Kumar and Varaiya 1986; Narendra and Annaswamy 1989; Sastry and Bodson 1989).The problem of flight control was, however, solved by using gain scheduling based on air data sensors and not by adaptive control (Stein 1980). The MRAC was also extended to nonlinear systems using *back-stepping* (Krstić et al. 1993); Lyapunov stability and passivity were essential ingredients in developing the algorithm and analyzing its stability.

The Self-Tuning Regulator

The self-tuning regulator was inspired by steady-state regulation in process control. The mathematical setting was discrete time stochastic systems. A block diagram of a system with a self-tuning regulator is shown in Fig. 2. The system has an ordinary feedback loop with a controller and the process. There is an external loop for adjusting the controller parameters based on real-time parameter estimation and control design. There are many ways to estimate the process parameters and many ways to do the control design. Simple schemes do not take parameter uncertainty into account when computing the controller parameters invoking *the certainty equivalence principle*.

Single-input, single-output stochastic systems can be modeled by

History of Adaptive Control, Fig. 2 Block diagram of a feedback system with a self-tuning regulator (STR)
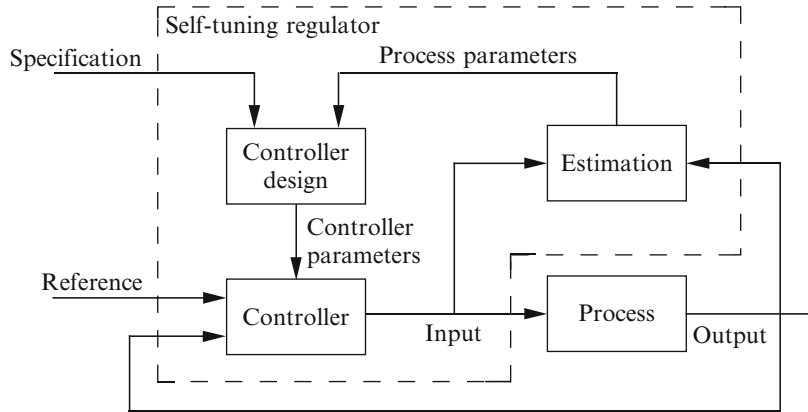
$$y(t) + a_1 y(t-h) + \cdots a_n y(t-nh) =$$
$$b_1 u(t-h) + \cdots + b_n u(t-nh) +$$
$$c_1 w(t-h) + \cdots c_n w(t-nh) + e(t), \tag{2}$$

where u is the control signal, y the process output, w a measured disturbance, and e a stochastic disturbance. Furthermore, h is the sampling period and a_k, b_k and c_k, are the parameters. Parameter estimation is typically done using least squares, and a control design that minimized the variance of the variations was well suited for regulation. A surprising result was that if the estimates converge, the limiting controller is a minimum variance controller even if the disturbance e is colored noise (Åström and Wittenmark 1973). Convergence conditions for the self-tuning regulator were given in Goodwin et al. (1980), and a very detailed analysis was presented in Guo and Chen (1991).

The problem of output feedback does not appear for the model (2) because the sequence of past inputs and outputs $y(t-h), \ldots, y(t-nh), u(t-h), \ldots, u(t-nh)$ is indeed a state, albeit not a minimal state representation. The continuous analogue would be to use derivatives of states and inputs which is not feasible because of measurement noise. The selection of the sampling period is however important.

Early industrial experience indicated that the ability of the STR to adapt feedforward gains was particularly useful, because feedforward control requires good models.

Insight from system identification showed that *excitation* is required to obtain good estimates. In the absence of excitation, a phenomenon of *bursting* could be observed. There could be epochs with small control actions due to insufficient excitation. The estimated parameters then drifted towards values close to or beyond the stability boundary generating large control axions. Good parameter estimates were then obtained and the system quickly recovered stability. The behavior then repeated in an irregular fashion. There are two ways to deal with the problem. One possibility is to detect when there is poor excitation and stop adaptation (Hägglund and Åström 2000). The other is to inject perturbations when there is poor excitation in the spirit of dual control.

Robustness and Unification

The model reference adaptive control and the self-tuning regulator originate from different application domains, flight control and process control. The differences are amplified because they are typically presented in different frameworks, continuous time for MRAC and discrete time for the STR. The schemes are, however, not too different. For a given process model and given design criterion the process model can often be re-parameterized in terms of controller parameters, and the STR is then equivalent to an MRAC. Similarly there are indirect MRAC where the process parameters are estimated (Egardt 1979).

A fundamental assumption made in the early analyses of model reference adaptive controllers was that the process model used for analysis had the same structure as the real process. Rohrs at MIT, which showed that systems with guaranteed convergence could be very sensitive to unmodeled dynamics, generated a good deal of research to explore robustness to unmodeled dynamics. Averaging theory, which is based on the observation that there are two loops in an adaptive system, a fast ordinary feedback and a slow parameter adjustment loop, turned out to be a key tool for understanding the behavior of adaptive systems. A large body of theory was generated and many books were written (Ioannou and Sun 1995; Sastry and Bodson 1989).

The theory resulted in several improvements of the adaptive algorithms. In the MIT rule (1) and similar adaptation laws derived from Lyapunov theory, the rate of change of the adaptation rate is a multiplication of the error e with other signals in the system. The adaptation rate may then become very large when signals are large. The analysis of robustness showed that there were advantages in avoiding large adaptation rates by *normalizing* the signals. The stability analysis also required that parameter estimates had to be bounded. To achieve this, parameters were *projected* on regions given by prior parameter bounds. The projection did, however, require prior process knowledge. The improved insight obtained from the robustness analysis is well described in the books Goodwin and Sin (1984), Egardt (1979), Åström and Wittenmark (1989), Narendra and Annaswamy (1989), Sastry and Bodson (1989), Anderson et al. (1986), and Ioannou and Sun (1995).

Applications

There were severe practical difficulties in implementing the early adaptive controllers using the analogue technology available in the brave era. Kalman used a hybrid computer when he attempted to implement his controller. There were dramatic improvements when mini- and microcomputers appeared in the 1970s. Since computers were still slow at the time, it was natural that most experimentats were executed in process control or ship steering which are slow processes. Advances in computing eliminated the technological barriers rapidly.

Self-oscillating adaptive controllers are used in several missiles. In piloted aircrafts there were complaints about the perturbation signals that were always exciting the system.

Self-tuning regulators have been used industrially since the early 1970s. Adaptive autopilots for ship steering were developed at the same time. They outperformed conventional autopilots based on PID control, because disturbances generated by waves were estimated and compensated for. These autopilots are still on the market (Northrop Grumman 2005). Asea (now ABB) developed a small distributed control system, Novatune, which had blocks for self-tuning regulators based on least-squares estimation, and minimum variance control. The company First Control, formed by members of the Novatune team, has delivered SCADA systems with adaptive control since 1985. The controllers are used for high-performance process control systems for pulp mills, paper machines, rolling mills, and pilot plants for chemical process control. The adaptive controllers are based on recursive estimation of a transfer function model and a control law based on pole placement. The controller also admits feedforward. The algorithm is provided with extensive safety logic, parameters are projected, and adaptation is interrupted when variations in measured signals and control signals are too small.

The most common industrial uses of adaptive techniques are automatic tuning of PID controllers. The techniques are used both in single loop controllers and in DCS systems. Many different techniques are used, pattern recognition as well as parameter estimation. The relay autotuning has proven very useful and has been shown to be very robust because it provides proper excitation of the process automatically. Some of the systems use automatic tuning to automatically generate gain schedules, and they also have adaptation of feedback and feedforward gains (Åström and Hägglund 2005).

Summary and Future Directions

Adaptive control has had turbulent history with alternating periods of optimism and pessimism. This history is reflected in the conferences. When the IEEE Conference on Decision and Control started in 1962, it included a Symposium on Adaptive Processes, which was discontinued after the 20th CDC in 1981. There were two IFAC symposia on the Theory of Self-Adaptive Control Systems, the first in Rome in 1962 and the second in Teddington in 1965 (Hammond 1966). The symposia were discontinued but reappeared when the Theory Committee of IFAC created a working group on adaptive control chaired by Prof. Landau in 1981. The group brought the communities of control and signal processing together, and a workshop on Adaptation and Learning in Signal Processing and Control (ALCOSP) was created. The first symposium was held in San Francisco in 1983 and the 11th in Caen in 2013.

Adaptive control can give significant benefits, it can deliver good performance over wide operating ranges, and commissioning of controllers can be simplified. Automatic tuning of PID controllers is now widely used in the process industry. Auto-tuning of more general controller is clearly of interest. Regulation performance is often characterized by the Harris index which compares actual performance with minimum variance control. Evaluation can be dispensed with by applying a self-tuning regulator.

There are adaptive controllers that have been in operation for more than 30 years, for example, in ship steering and rolling mills. There is a variety of products that use scheduling, MRAC, and STR in different ways. Automatic tuning is widely used; virtually all new single loop controllers have some form of automatic tuning. Automatic tuning is also used to build gain schedules semiautomatically. The techniques appear in tuning devices, in single loop controllers, in distributed systems for process control, and in controllers for special applications. There are strong similarities between adaptive filtering and adaptive control. Noise cancellation and adaptive equalization are widely spread uses of adaptation. The signal processing applications are a little easier to analyze because the systems do not have a feedback controller. New adaptive schemes are appearing. The $\mathcal{L}_1$ adaptive controller is one example. It inherits features of both the STR and the MRAC. The *model-free controller* by Fliess and Join (2013) is another example. It is similar to a continuous time version of the self-tuning regulator.

There is renewed interest in adaptive control in the aerospace industry, both for aircrafts and missiles (Lavretsky and Wise 2013). Good results in flight tests have been reported both using MRAC and the recently developed $\mathcal{L}_1$ adaptive controller (Hovakimyan and Cao 2010).

Adaptive control is a rich field, and to understand it well, it is necessary to know a wide range of techniques: nonlinear, stochastic, and sampled data systems, stability, robust control, and system identification.

In the early development of adaptive control, there was a dream of the universal adaptive controller that could be applied to any process with very little prior process knowledge. The insight gained by the robustness analysis shows that knowledge of bounds on the parameters is essential to ensure robustness. With the knowledge available today, adaptive controllers can be designed for particular applications. Design of proper safety nets is an important practical issue. One useful approach is to start with a basic constant-gain controller and provide adaptation as an add-on. This approach also simplifies design of supervision and safety networks.

There are still many unsolved research problems. Methods to determine the achievable adaptation rates are not known. Finding ways to provide proper excitation is another problem. The dual control formulation is very attractive because it automatically generates proper excitation when it is needed. The computations required to solve the Bellman equations are prohibitive, except in very simple cases. The self-oscillating adaptive system, which has been successfully applied to missiles, does provide excitation. The success of the relay auto-tuner for simple controllers indicates that it may be called in to provide excitation of adaptive controllers. Adaptive control can be an important

component of the emerging autonomous system. One may expect that the current upswing in systems biology may provide more inspiration because many biological clearly have adaptive capabilities.

Cross-References

▶ Adaptive Control, Overview
▶ Autotuning
▶ Extremum Seeking Control
▶ Model Reference Adaptive Control
▶ PID Control

Bibliography

Anderson BDO, Bitmead RR, Johnson CR, Kokotović PV, Kosut RL, Mareels I, Praly L, Riedle B (1986) Stability of adaptive systems. MIT, Cambridge

Åström KJ, Hägglund T (2005) Advanced PID control. ISA – The Instrumentation, Systems, and Automation Society, Research Triangle Park

Åström KJ, Wittenmark B (1973) On self-tuning regulators. Automatica 9:185–199

Åström KJ, Wittenmark B (1989) Adaptive control. Addison-Wesley, Reading. Second 1994 edition reprinted by Dover 2006 edition

Bellman R (1961) Adaptive control processes—a guided tour. Princeton University Press, Princeton

Butchart RL, Shackcloth B (1965) Synthesis of model reference adaptive control systems by Lyapunov's second method. In: Proceedings 1965 IFAC symposium on adaptive control, Teddington

Draper CS, Li YT (1966) Principles of optimalizing control systems and an application to the internal combustion engine. In: Oldenburger R (ed) Optimal and self-optimizing control. MIT, Cambridge

Egardt B (1979) Stability of adaptive controllers. Springer, Berlin

Fliess M, Join C (2013) Model-free control.

Foxboro (1950) Control system with automatic response adjustment. US Patent 2,517,081

Gabor D, Wilby WPL, Woodcock R (1959) A universal non-linear filter, predictor and simulator which optimizes itself by a learning process. Proc Inst Electron Eng 108(Part B):1061–, 1959

Goodwin GC, Sin KS (eds) (1984) Adaptive filtering prediction and control. Prentice Hall, Englewood Cliffs

Goodwin GC, Ramadge PJ, Caines PE (1980) Discrete-time multivariable adaptive control. IEEE Trans Autom Control AC-25:449–456

Gregory PC (ed) (1959) Proceedings of the self adaptive flight control symposium. Wright Air Development Center, Wright-Patterson Air Force Base, Ohio

Guo L, Chen HF (1991) The Åström-Wittenmark's self-tuning regulator revisited and ELS-based adaptive trackers. IEEE Trans Autom Control 30(7):802–812

Hägglund T, Åström KJ (2000) Supervision of adaptive control algorithms. Automatica 36:1171–1180

Hammond PH (ed) (1966) Theory of self-adaptive control systems. In: Proceedings of the second IFAC symposium on the theory of self-adaptive control systems, 14–17 Sept, National Physical Laboratory, Teddington. Plenum, New York

Hebb DO (1949) The organization of behavior. Wiley, New York

Hovakimyan N, Chengyu Cao (2010) $\mathcal{L}_1$ adaptive control theory. SIAM, Philadelphia

Ioannou PA, Sun J (1995) Stable and robust adaptive control. Prentice-Hall, Englewood Cliffs

Kalman RE (1958) Design of a self-optimizing control system. Trans ASME 80:468–478

Krstić M, Kanellakopoulos I, Kokotović PV (1993) Non-linear and adaptive control design. Prentice Hall, Englewood Cliffs

Kumar PR, Varaiya PP (1986) Stochastic systems: estimation, identification and adaptive control. Prentice-Hall, Englewood Cliffs

Landau ID (1979) Adaptive control—the model reference approach. Marcel Dekker, New York

Lavretsky E, Wise KA (2013) Robust and adaptive control with aerospace applications. Springer, London

Mishkin E, Braun L (1961) Adaptive control systems. McGraw-Hill, New York

Monopoli RV (1974) Model reference adaptive control with an augmented error signal. IEEE Trans Autom Control AC-19:474–484

Morse AS (1980) Global stability of parameter-adaptive control systems. IEEE Trans Autom Control AC-25:433–439

Narendra KS, Annaswamy AM (1989) Stable adaptive systems. Prentice Hall, Englewook Cliffs

Northrop Grumman (2005) SteerMaster. http://www.srhmar.com/brochures/as/SPERRY%20SteerMaster%20Control%20System.pdf

Parks PC (1966) Lyapunov redesign of model reference adaptive control systems. IEEE Trans Autom Control AC-11:362–367

Sastry S, Bodson M (1989) Adaptive control: stability, convergence and robustness. Prentice-Hall, New Jersey

Simon HA (1956) Dynamic programming under uncertainty with a quadratic criterion function. Econometrica 24:74–81

Stein G (1980) Adaptive flight control: a pragmatic view. In: Narendra KS, Monopoli RV (eds) Applications of adaptive control. Academic, New York

Tsypkin YaZ (1971) Adaptation and learning in automatic systems. Academic, New York

Widrow B (1962) Generalization and information storage in network of Adaline neurons. In: Yovits et al. (ed) Self-organizing systems. Spartan Books, Washington

Hybrid Dynamical Systems, Feedback Control of

Ricardo G. Sanfelice
Department of Computer Engineering,
University of California at Santa Cruz,
Santa Cruz, CA, USA

Abstract

The control of systems with hybrid dynamics requires algorithms capable of dealing with the intricate combination of continuous and discrete behavior, which typically emerges from the presence of continuous processes, switching devices, and logic for control. Several analysis and design techniques have been proposed for the control of nonlinear continuous-time plants, but little is known about controlling plants that feature truly hybrid behavior. This short entry focuses on recent advances in the design of feedback control algorithms for hybrid dynamical systems. The focus is on hybrid feedback controllers that are systematically designed employing Lyapunov-based methods. The control design techniques summarized in this entry include control Lyapunov function-based control, passivity-based control, and trajectory tracking control.

Keywords

Feedback control; Hybrid control; Hybrid systems; Asymptotic stability

Definition

A *hybrid control system* is a *feedback system* whose variables may flow and, at times, jump. Such a hybrid behavior can be present in one or more of the subsystems of the feedback system: in the system to control, i.e., *the plant*; in the algorithm used for control, i.e., *the controller*; or in the subsystems needed to interconnect the plant and the controller, i.e., *the interfaces/signal conditioners*. Figure 1 depicts a feedback system in closed-loop configuration with such subsystems under the presence of environmental disturbances. Due to its hybrid dynamics, a hybrid control system is a particular type of *hybrid dynamical system*.

Motivation

Hybrid dynamical systems are ubiquitous in science and engineering as they permit capturing the complex and intertwined continuous/discrete behavior of a myriad of systems with variables that flow and jump. The recent popularity of feedback systems combining physical and software components demands tools for stability analysis and control design that can systematically handle such a complex combination. To avoid the issues due to approximating the dynamics of a system, in numerous settings, it is mandatory to keep the system dynamics as pure as possible and to be able to design feedback controllers that can cope with flow and jump behavior in the system.

Modeling Hybrid Dynamical Control Systems

In this entry, hybrid control systems are represented in the framework of *hybrid equations/inclusions* for the study of hybrid dynamical systems. Within this framework, the continuous dynamics of the system are modeled using a differential equation/inclusion, while the discrete dynamics are captured by a difference equation/inclusion. A solution to such a system can *flow* over nontrivial intervals of time and *jump* at certain time instants. The conditions determining whether a solution to a hybrid system should flow or jump are captured by subsets of the state space and input space of the hybrid control system. In this way, a *plant* with hybrid dynamics can be modeled by the hybrid inclusion.
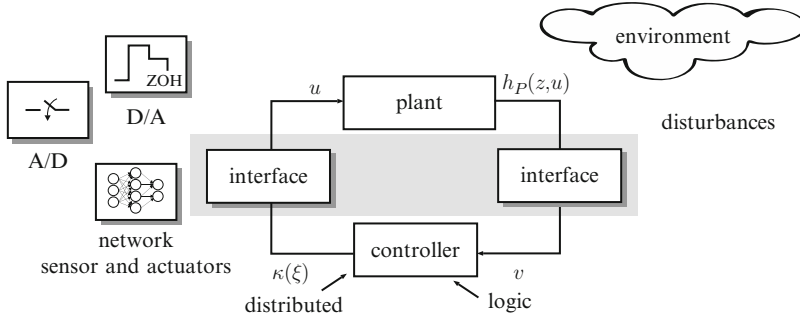
Hybrid Dynamical Systems, Feedback Control of, Fig. 1 A hybrid control system: a feedback system with a plant, controller, and interfaces/signal conditioners

(along with environmental disturbances) as subsystems featuring variables that flow and, at times, jump

$$E \cap ([0, T] \times \{0, 1, \dots J\})$$

can be written in the form

$$\bigcup_{j=0}^{J-1} \left([t_j, t_{j+1}], j \right)$$

for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_J$. A hybrid arc ϕ is a function on a hybrid time domain. The set $E \cap ([0, T] \times \{0, 1, \dots, J\})$ defines a compact hybrid time domain since it is bounded and closed. The hybrid time domain of ϕ is denoted by $\mathrm{dom}\,\phi$. A hybrid arc is such that, for each $j \in \mathbb{N}$, $t \mapsto \phi(t, j)$ is absolutely continuous on intervals of flow $I^j := \{t : (t, j) \in \mathrm{dom}\,\phi\}$ with nonzero Lebesgue measure. A hybrid input u is a function on a hybrid time domain that, for each $j \in \mathbb{N}$, $t \mapsto u(t, j)$ is Lebesgue measurable and locally essentially bounded on the interval I^j.

In this way, a solution to the plant $\mathcal{H}_P$ is given by a pair (ϕ, u) with $\mathrm{dom}\,\phi = \mathrm{dom}\,u$ ($= \mathrm{dom}(\phi, u)$) satisfying

(S0) $(\phi(0,0), u(0,0)) \in \overline{C}_P$ *or* $(\phi(0,0), u(0,0)) \in D_P$, *and* $\mathrm{dom}\,\phi = \mathrm{dom}\,u$;

(S1) For each $j \in \mathbb{N}$ such that I^j has nonempty interior $\mathrm{int}(I^j)$*, we have*

$$(\phi(t, j), u(t, j)) \in C_P \quad \text{for all } t \in \mathrm{int}(I^j)$$

and

$$\frac{d}{dt}\phi(t, j) \in F_P(\phi(t, j), u(t, j))$$
$$\text{for almost all } t \in I^j$$

$$\mathcal{H}_P : \begin{cases} \dot{z} \in F_P(z, u) & (z, u) \in C_P \\ z^+ \in G_P(z, u) & (z, u) \in D_P \\ y = h_P(z, u) \end{cases} \quad (1)$$

where z is the *state* of the plant and takes values from the Euclidean space $\mathbb{R}^{n_P}$, u is the *input* and takes values from $\mathbb{R}^{m_P}$, y is the *output* and takes values from the output space $\mathbb{R}^{r_P}$, and $(C_P, F_P, D_P, G_P, h_P)$ is the *data* of the hybrid system. The set C_P is the *flow set*, the set-valued map F_P is the *flow map*, the set D_P is the *jump set*, the set-valued map G_P is the *jump map*, and the single-valued map h_P is the *output map*. (This hybrid inclusion captures the dynamics of (constrained or unconstrained) continuous-time systems when $D_P = \emptyset$ and G_P is arbitrary. Similarly, it captures the dynamics of (constrained or unconstrained) discrete-time systems when $C_P = \emptyset$ and F_P is arbitrary. Note that while the output inclusion does not explicitly include a constraint on (z, u), the output map is only evaluated along solutions.)

Given an input u, a *solution to a hybrid inclusion* is defined by a state trajectory ϕ that satisfies the inclusions. Both the input and the state trajectory are functions of $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N} := [0, \infty) \times \{0, 1, 2, \dots\}$, where t keeps track of the amount of flow, while j counts the number of jumps of the solution. These functions are given by *hybrid arcs* and *hybrid inputs*, which are defined on *hybrid time domains*. More precisely, hybrid time domains are subsets E of $\mathbb{R}_{\geq 0} \times \mathbb{N}$ that, for each $(T, J) \in E$,

(S2) For each $(t, j) \in \text{dom}(\phi, u)$ *such that* $(t, j + 1) \in \text{dom}(\phi, u)$, *we have*

$$(\phi(t, j), u(t, j)) \in D_P$$

and

$$\phi(t, j + 1) \in G_P(\phi(t, j), u(t, j))$$

A solution pair (ϕ, u) to $\mathcal{H}$ is said to be *complete* if $\text{dom}(\phi, u)$ is unbounded and *maximal* if there does not exist another pair $(\phi, u)'$ such that (ϕ, u) is a truncation of $(\phi, u)'$ to some proper subset of $\text{dom}(\phi, u)'$. A solution pair (ϕ, u) to $\mathcal{H}$ is said to be *Zeno* if it is complete and the projection of $\text{dom}(\phi, u)$ onto $\mathbb{R}_{\geq 0}$ is bounded.

Input and output modeling remark: At times, it is convenient to define inputs $u_c \in \mathbb{R}^{m_{P,c}}$ and $u_d \in \mathbb{R}^{m_{P,d}}$ collecting every component of the input u that affect flows and that affect jumps, respectively (Some of the components of u can be used to define both u_c and u_d, that is, there could be inputs that affect both flows and jumps.). Similarly, one can define y_c and y_d as the components of y that are measured during flows and jumps, respectively.

To control the hybrid plant $\mathcal{H}_P$ in (1), control algorithms that can cope with the nonlinearities introduced by the flow and jump equations/inclusions are required. In general, feedback controllers designed using classical techniques from the continuous-time and discrete-time domain fall short. Due to this limitation, hybrid feedback controllers would be more suitable for the control of plants with hybrid dynamics. Then, following the hybrid plant model above, hybrid controllers for the plant $\mathcal{H}_P$ in (1) will be given by the hybrid inclusion

$$\mathcal{H}_K \ : \ \begin{cases} \dot{\xi} \ \in \ F_K(\xi, v) & (\xi, v) \in C_K \\ \xi^+ \in \ G_K(\xi, v) & (\xi, v) \in D_K \\ \eta \ = \ \kappa(\xi, v) \end{cases} \quad (2)$$

where ξ is the *state* of the controller and takes values from the Euclidean space $\mathbb{R}^{n_K}$, v is the *input* and takes values from $\mathbb{R}^{r_P}$, η is the *output* and takes values from the output space $\mathbb{R}^{m_P}$, and

$(C_K, F_K, D_K, G_K, \kappa)$ is the *data* of the hybrid inclusion defining the hybrid controller.

The control of $\mathcal{H}_P$ via $\mathcal{H}_K$ defines an interconnection through the input/output assignment $u = \eta$ and $v = y$; the system in Fig. 1 without interfaces represents this interconnection. The resulting closed-loop system is a hybrid dynamical system given in terms of a hybrid inclusion/equation with state $x = (z, \xi)$. We will denote such a closed-loop system by $\mathcal{H}$. Its data can be constructed from the data $(C_P, F_P, D_P, G_P, h_P)$ and $(C_K, F_K, D_K, G_K, \kappa)$ of each of the subsystems. Solutions to both $\mathcal{H}_K$ and $\mathcal{H}$ are understood following the notion introduced above.

Definitions and Notions

For convenience, we use the equivalent notation $[x^\top \ y^\top]^\top$ and (x, y) for vectors x and y. Also, we denote by $\mathcal{K}_\infty$ the class of functions from $\mathbb{R}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ that are continuous, zero at zero, strictly increasing, and unbounded.

The dynamics of hybrid inclusions have right-hand sides given by set-valued maps. Unlike functions or single-valued maps, set-valued maps may return a set when evaluated at a point. For instance, at points in C_P, the set-valued flow map F_P of the hybrid plant $\mathcal{H}_P$ might return more than one value, allowing for different values of the derivative of z. A particular continuity property of set-valued maps that will be needed later is lower semicontinuity. A set-valued map S from $\mathbb{R}^n$ to $\mathbb{R}^m$ is lower semicontinuous if for each $x \in \mathbb{R}^n$ one has that $\liminf_{x_i \to x} S(x_i) \supset S(x)$, where $\liminf_{x_i \to x} S(x_i) = \{z \ : \ \forall x_i \to x, \exists z_i \to z$ s.t. $z_i \in S(x_i) \}$ is the so-called *inner limit* of S.

A vast majority of control problems consist of designing a feedback algorithm that assures that a function of the solutions to the plant approach a desired set-point condition (*attractivity*) and, when close to it, the solutions remain nearby (*stability*). In some scenarios, the desired set-point condition is not necessarily an isolated point, but rather a set. The problem of designing a hybrid controller $\mathcal{H}_K$ for a hybrid plant $\mathcal{H}_P$ typically pertains to the stabilization of sets, in

particular, due to the hybrid controller's state including timers that persistently evolve within a bounded time interval and logic variables that take values from discrete sets. Denoting by $\mathcal{A}$ the set of points to stabilize for the closed-loop system $\mathcal{H}$ and $|\cdot|_{\mathcal{A}}$ as the distance to such set, the following property captures the typically desired properties outlined above. A closed set $\mathcal{A}$ is said to be:

(S) *Stable*: for each $\varepsilon > 0$ there exists $\delta > 0$ such that each maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_{\circ}$, $|x_{\circ}|_{\mathcal{A}} \leq \delta$ satisfies $|\phi(t, j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t, j) \in \mathrm{dom}\,\phi$.

(A) *Attractive*: there exists $\mu > 0$ such that every maximal solution ϕ to $\mathcal{H}$ with $\phi(0, 0) = x_{\circ}$, $|x_{\circ}|_{\mathcal{A}} \leq \mu$ is bounded and if it is complete satisfies $\lim_{(t, j) \in \mathrm{dom}\,\phi,\ t+j \to \infty} |\phi(t, j)|_{\mathcal{A}} = 0$.

(AS) *Asymptotically stable*: it is stable and attractive.

The basin of attraction of an asymptotically stable set $\mathcal{A}$ is the set of points from where the attractivity property holds. The set $\mathcal{A}$ is said to be globally asymptotically stable when the basin of attraction is equal to the entire state space.

A dynamical system with assigned inputs is said to be detectable when its output being held to zero implies that its state converges to the origin. A similar property can be defined for hybrid dynamical systems. For the closed-loop system $\mathcal{H}$, given sets $\mathcal{A}$ and K, the distance to $\mathcal{A}$ is 0-input detectable relative to K for $\mathcal{H}$ if every complete solution ϕ to $\mathcal{H}$

$$\phi(t, j) \in K \qquad \forall (t, j) \in \mathrm{dom}\,\phi \qquad \Rightarrow$$

$$\lim_{(t, j) \in \mathrm{dom}\,\phi,\ t+j \to \infty} |\phi(t, j)|_{\mathcal{A}} = 0$$

where "$\phi(t, j) \in K$" captures the "output being held to zero" property in the usual detectability notion.

Feedback Control Design for Hybrid Dynamical Systems

Several methods for the design of a hybrid controller $\mathcal{H}_K$ rendering a given set asymptotically stable are given below. At the core of these methods are sufficient conditions in terms of Lyapunov functions guaranteeing that the asymptotic stability property defined in section "Definitions and Notions" holds. Some of the methods presented below exploit such sufficient conditions when applied to the closed-loop system $\mathcal{H}$, while others exploit the properties of the hybrid plant to design controllers with a particular structure. The design methods are presented in order of complexity of the controller, namely, from it being a static state-feedback law to being a generic algorithm with true hybrid dynamics.

CLF-Based Control Design

In simple terms, a control Lyapunov function (CLF) is a regular enough scalar function that decreases along solutions to the system for some values of the unassigned input. When such a function exists, it is very tempting to exploit its properties to construct an asymptotically stabilizing control law. Following the ideas from the literature of continuous-time and discrete-time nonlinear systems, we define control Lyapunov functions for hybrid plants $\mathcal{H}_P$ and present results on CLF-based control design. For simplicity, as mentioned in the *input and output modeling remark* in section "Definitions and Notions," we use inputs u_c and u_d instead u. Also, we restrict the discussion to sets $\mathcal{A}$ that are compact as well as hybrid plants with F_P, G_P single valued and such that $h_P(z, u) = z$. For notational convenience, we use Π to denote the "projection" of C_P and D_P onto $\mathbb{R}^{n_P}$, i.e., $\Pi(C_P) = \{z : \exists u_c \text{ s.t. } (z, u_c) \in C_P\}$ and $\Pi(D_P) = \{z : \exists u_d \text{ s.t. } (z, u_d) \in D_P\}$, and the set-valued maps $\Psi_c(z) = \{u_c : (z, u_c) \in C_P\}$ and $\Psi_d(z) = \{u_d : (z, u_d) \in D_P\}$.

Given a compact set $\mathcal{A}$, a continuously differentiable function $V : \mathbb{R}^{n_P} \to \mathbb{R}$ is a *control Lyapunov function for $\mathcal{H}_P$ with respect to $\mathcal{A}$* if there exist $\alpha_1, \alpha_2 \in \mathcal{K}_{\infty}$ and a continuous, positive definite function ρ such that

$$\alpha_1\,(|z|_{\mathcal{A}}) \leq V(z) \leq \alpha_2(|z|_{\mathcal{A}})$$
$$\forall z \in \mathbb{R}^{n_P}$$
$$\inf_{u_c \in \Psi_c(z)} \langle \nabla V(z), F_P(z, u_c) \rangle \leq -\rho(|z|_{\mathcal{A}})$$

$$\forall z \in \Pi(C_P) \qquad (3)$$

$$\inf_{u_d \in \Psi_d(z)} V(G_P(z, u_d)) - V(z) \leq -\rho(|z|_{\mathcal{A}})$$

$$\forall z \in \Pi(D_P) \qquad (4)$$

With the availability of a CLF, the set $\mathcal{A}$ can be asymptotically stabilized if it is possible to synthesize a controller $\mathcal{H}_K$ from inequalities (3) and (4). Such a synthesis is feasible, in particular, forthe special case of $\mathcal{H}_K$ being a

static state-feedback law $z \mapsto \kappa(z)$. Sufficient conditions guaranteeing the existence of such a controller as well as a particular state-feedback law with point-wise minimum norm are given next.

Given a compact set $\mathcal{A}$ and a control Lyapunov function V (with respect to $\mathcal{A}$), define, for each $r \geq 0$, the set $\mathcal{I}(r) := \{z \in \mathbb{R}^{n_P} : V(z) \geq r\}$. Moreover, for each (z, u_c) and $r \geq 0$, define the function

$$\Gamma_c(z, u_c, r) := \begin{cases} \langle \nabla V(z), F_P(z, u_c) \rangle + \frac{1}{2}\rho(|z|_{\mathcal{A}}) & \text{if } (z, u_c) \in C_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_{P,c}}), \\ -\infty & \text{otherwise} \end{cases}$$

and, for each (z, u_d) and $r \geq 0$, the function

$$\Gamma_d(z, u_d, r) := \begin{cases} V(G_P(z, u_d)) - V(z) + \frac{1}{2}\rho(|z|_{\mathcal{A}}) & \text{if } (z, u_d) \in D_P \cap (\mathcal{I}(r) \times \mathbb{R}^{m_{P,d}}), \\ -\infty & \text{otherwise} \end{cases}$$

The following result states conditions on the data of $\mathcal{H}_P$ guaranteeing that, for each $r > 0$, there exists a continuous state-feedback law $z \mapsto \kappa(z) = (\kappa_c(z), \kappa_d(z))$ rendering the compact set

$$\mathcal{A}_r := \{z \in \mathbb{R}^{n_P} : V(z) \leq r\}$$

asymptotically stable. This property corresponds to a practical version of asymptotic stabilizability.

Theorem 1 *Given a hybrid plant $\mathcal{H}_P = (C_P, F_P, D_P, G_P, h_P)$, a compact set $\mathcal{A}$, and a control Lyapunov function V for $\mathcal{H}_P$ with respect to $\mathcal{A}$, if*
(C1) C_P and D_P are closed sets, and F_P and G_P are continuous;
(C2) The set-valued maps $\Psi_c(z) = \{u_c : (z, u_c) \in C_P\}$ and $\Psi_d(z) = \{u_d : (z, u_d) \in D_P\}$ are lower semicontinuous with convex values;
(C3) For every $r > 0$, we have that, for every $z \in \Pi(C_P) \cap \mathcal{I}(r)$, the function $u_c \mapsto \Gamma_c(z, u_c, r)$ is convex on $\Psi_c(z)$ and that, for every $z \in \Pi(D_P) \cap \mathcal{I}(r)$, the function $u_d \mapsto \Gamma_c(z, u_d, r)$ is convex on $\Psi_d(z)$;
then, for every $r > 0$, the compact set $\mathcal{A}_r$ is asymptotically stabilizable for $\mathcal{H}_P$ by a state-

feedback law $z \mapsto \kappa(z) = (\kappa_c(z), \kappa_d(z))$ with κ_c continuous on $\Pi(C_P) \cap \mathcal{I}(r)$ and κ_d continuous on $\Pi(D_P) \cap \mathcal{I}(r)$.

Theorem 1 assures the existence of a continuous state-feedback law practically asymptotically stabilizing $\mathcal{A}$. However, Theorem 1 does not provide an expression of an asymptotically stabilizing control law. The following result provides an explicit construction of such a control law.

Theorem 2 *Given a hybrid plant $\mathcal{H}_P = (C_P, F_P, D_P, G_P, h_P)$, a compact set $\mathcal{A}$, and a control Lyapunov function V for $\mathcal{H}_P$ with respect to $\mathcal{A}$, if (C1)–(C3) in Theorem 1 hold then, for every $r > 0$, the state-feedback law pair*

$$\kappa_c : \Pi(C_P) \to \mathbb{R}^{m_{P,c}}, \quad \kappa_d : \Pi(D_P) \to \mathbb{R}^{m_{P,d}}$$

defined on $\Pi(C_P)$ and $\Pi(D_P)$ as

$$\kappa_c(z) := \arg\min\{|u_c| : u_c \in \mathcal{T}_c(z)\}$$

$$\forall z \in \Pi(C_P) \cap \mathcal{I}(r)$$

$$\kappa_d(z) := \arg\min\{|u_d| : u_d \in \mathcal{T}_d(z)\}$$

$$\forall z \in \Pi(D_P) \cap \mathcal{I}(r)$$

respectively, renders the compact set $\mathcal{A}_r$ asymptotically stable for $\mathcal{H}_P$, where $\mathcal{T}_c(z) = \Psi_c(z) \cap \{u_c : \Gamma_c(z, u_c, V(z)) \leq 0\}$ and $\mathcal{T}_d(z) = \Psi_d(z) \cap \{u_d : \Gamma_d(z, u_d, V(z)) \leq 0\}$. Furthermore, if the set-valued maps Ψ_c and Ψ_d have a closed graph, then κ_c and κ_d are continuous on $\Pi(C_P) \cap \mathcal{I}(r)$ and $\Pi(D_P) \cap \mathcal{I}(r)$, respectively.

The stability properties guaranteed by Theorems 1 and 2 are practical. Under further properties, similar results hold when the input u is not partitioned into u_c and u_d. To achieve asymptotic stability (or stabilizability) of $\mathcal{A}$ with a continuous state-feedback law, extra conditions are required to hold nearby the compact set, which for the case of stabilization of continuous-time systems are the so-called *small control properties*. Furthermore, the continuity of the feedback law assures that the closed-loop system has closed flow and jump sets as well as continuous flow and jump maps, which, in turn, due to the compactness of $\mathcal{A}$, implies that the asymptotic stability property is robust. Robustness follows from results for hybrid systems without inputs.

Passivity-Based Control Design

Dissipativity and its special case, passivity, provide a useful physical interpretation of a feedback control system as they characterize the exchange of energy between the plant and its controller. For an open system, passivity (in its very pure form) is the property that the energy stored in the system is no larger than the energy it has absorbed over a period of time. The energy stored in a system is given by the difference between the initial and final energy over a period of time, where the energy function is typically called the *storage function*. Hence, conveniently, passivity can be expressed in terms of the derivative of a storage function (i.e., the rate of change of the internal energy) and the product between inputs and outputs (i.e., the system's power flow). Under further observability conditions, this power inequality can be employed as a design tool by selecting a control law that makes the rate of change of the internal energy negative. This method is called *passivity-based control design*.

The passivity-based control design method can be employed in the design of a controller for

a "passive" hybrid plant $\mathcal{H}_P$, in which energy might be dissipated during flows, jumps, or both. Passivity notions and a passivity-based control design method for hybrid plants are given next. Since the form of the plant's output plays a key role in asserting a passivity property, and this property may not necessarily hold both during flows and jumps, as mentioned in the *input and output modeling remark* in section "Definitions and Notions," we define outputs y_c and y_d, which, for simplicity, are assumed to be single valued: $y_c = h_c(x)$ and $y_d = h_d(x)$. Moreover, we consider the case when the dimension of the space of the inputs u_c and u_d coincides with that of the outputs y_c and y_d, respectively, i.e., a "duality" of the output and input space.

Given a compact set $\mathcal{A}$ and functions h_c, h_d such that $h_c(\mathcal{A}) = h_d(\mathcal{A}) = 0$, a hybrid plant $\mathcal{H}_P$ for which there exists a continuously differentiable function $V : \mathbb{R}^{n_P} \rightarrow \mathbb{R}_{\geq 0}$ satisfying for some functions $\omega_c : \mathbb{R}^{m_{P,c}} \times \mathbb{R}^{n_P} \rightarrow \mathbb{R}$ and $\omega_d : \mathbb{R}^{m_{P,c}} \times \mathbb{R}^{n_P} \rightarrow \mathbb{R}$

$$\langle \nabla V(z), F_P(z, u_c) \rangle \leq \omega_c(u_c, z)$$
$$\forall (z, u_c) \in C \tag{5}$$
$$V(G_P(z, u_d)) - V(z) \leq \omega_d(u_d, z)$$
$$\forall (z, u_d) \in D \tag{6}$$

is said to be *passive with respect to a compact set* $\mathcal{A}$ if

$$(u_c, z) \mapsto \omega_c(u_c, z) = u_c^\top y_c \tag{7}$$
$$(u_d, z) \mapsto \omega_d(u_d, z) = u_d^\top y_d \tag{8}$$

The function V is the so-called *storage function*. If (5) holds with ω_c as in (7), and (6) holds with $\omega_d \equiv 0$, then the system is called *flow-passive*, i.e., the power inequality holds only during flows. If (5) holds with $\omega_c \equiv 0$, and (6) holds with ω_d as in (8), then the system is called *jump-passive*, i.e., the energy of the system decreases only during jumps.

Under additional detectability properties, these passivity notions can be used to design static output feedback controllers. The following result gives two design methods for hybrid plants.

Theorem 3 *Given a hybrid plant* $\mathcal{H}_P = (C_P, F_P, D_P, G_P, h_P)$ *satisfying*

(C1′) C_P *and* D_P *are closed sets;* F_P *and* G_P *are continuous; and* h_c *and* h_d *are continuous; and a compact set* $\mathcal{A}$, *we have:*

(1) If $\mathcal{H}_P$ *is flow-passive with respect to* $\mathcal{A}$ *with a storage function* V *that is positive definite with respect to* $\mathcal{A}$ *and has compact sublevel sets, and if there exists a continuous function* $\kappa_c : \mathbb{R}^{m_{P,c}} \to \mathbb{R}^{m_{P,c}}$, $y_c^\top \kappa_c(y_c) > 0$ *for all* $y_c \neq 0$, *such that the resulting closed-loop system with* $u_c = -\kappa_c(y_c)$ *and* $u_d \equiv 0$ *has the following properties:*

(1.1) The distance to $\mathcal{A}$ *is detectable relative to*

$$\{z \in \Pi(C_P) \cup \Pi(D_P) \cup G_P(D_P) :$$
$$h_c(z)^\top \kappa_c(h_c(z)) = 0, (z, -\kappa_c(h_c(z))) \in C_P\};$$

(1.2) Every complete solution ϕ *is such that, for some* $\delta > 0$ *and some* $J \in \mathbb{N}$, *we have* $t_{j+1} - t_j \geq \delta$ *for all* $j \geq J$;
then the control law $u_c = -\kappa_c(y_c)$, $u_d \equiv 0$ *renders* $\mathcal{A}$ *globally asymptotically stable.*

(2) If $\mathcal{H}_P$ *is jump-passive with respect to* $\mathcal{A}$ *with a storage function* V *that is positive definite with respect to* $\mathcal{A}$ *and has compact sublevel sets, and if there exists a continuous function* $\kappa_d : \mathbb{R}^{m_{P,d}} \to \mathbb{R}^{m_{P,d}}$, $y_d^\top \kappa_d(y_d) > 0$ *for all* $y_d \neq 0$, *such that the resulting closed-loop system with* $u_c \equiv 0$ *and* $u_d = -\kappa_d(y_d)$ *has the following properties:*

(2.1) The distance to $\mathcal{A}$ *is detectable relative to*

$$\{z \in \Pi(C_P) \cup \Pi(D_P) \cup G_P(D_P) :$$
$$h_d(z)^\top \kappa_d(h_d(z)) = 0, (z, -\kappa_d(h_d(z))) \in D_P\};$$

(2.2) Every complete solution ϕ *is Zeno;*
then the control law $u_d = -\kappa_d(y_d)$, $u_c \equiv 0$ *renders* $\mathcal{A}$ *globally asymptotically stable.*

Strict passivity notions can also be formulated for hybrid plants, including the special cases where the power inequalities hold only during flows or jumps. In particular, strict passivity and output strict passivity can be employed to assert asymptotic stability with zero inputs.

Tracking Control Design

While numerous control problems pertain to the stabilization of a set-point condition, at times, it is desired to stabilize the solutions to the plant to a time-upying trajectory. In this section, we consider the problem of designing a hybrid controller $\mathcal{H}_K$ for a hybrid plant $\mathcal{H}_P$ to *track* a given reference trajectory r (a hybrid arc). The notion of tracking is introduced below. We propose sufficient conditions that general hybrid plants and controllers should satisfy to solve such a problem. For simplicity, we consider tracking of state trajectories and that the hybrid controller can measure both the state of the plant z and the reference trajectory r; hence, $v = (z, r)$.

The particular approach used here consists of recasting the tracking control problem as a set stabilization problem for the closed-loop system $\mathcal{H}$. To do this, we embed the reference trajectory r into an augmented hybrid model for which it is possible to define a set capturing the condition that the plant tracks the given reference trajectory. This set is referred to as *the tracking set*. More precisely, given a reference $r : \operatorname{dom} r \to \mathbb{R}^{n_p}$, we define the set $\mathcal{T}_r$ collecting all of the points (t, j) in the domain of r at which r jumps, that is, every point $(t_j^r, j) \in \operatorname{dom} r$ such that $(t_j^r, j + 1) \in \operatorname{dom} r$. Then, the state of the closed loop $\mathcal{H}$ is augmented by the addition of states $\tau \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{N}$. The dynamics of the states τ and k are such that τ counts elapsed flow time, while k counts the number of jumps of $\mathcal{H}$; hence, during flows $\dot{\tau} = 1$ and $\dot{k} = 0$, while at jumps $\tau^+ = \tau$ and $k^+ = k + 1$. These new states are used to parameterize the given reference trajectory r, which is employed in the definition of the tracking set

$$\mathcal{A} = \{(z, \eta, \tau, k) \in \mathbb{R}^{n_P} \times \mathbb{R}^{n_K} \times \mathbb{R}_{\geq 0} \times \mathbb{N} :$$
$$z = r(\tau, k), \xi \in \Phi_K\} \tag{9}$$

This set is the target set to be stabilized for $\mathcal{H}$. The set $\Phi_K \subset \mathbb{R}^{n_K}$ in the definition of $\mathcal{A}$ is some closed set capturing the set of points asymptotically approached by the controller's state ξ.

The following result establishes a sufficient condition for stabilization of the tracking set

$\mathcal{A}$. For notational convenience, we define $x = (z, \xi, \tau, k)$,

$$C = \{x \,:\, (z, \kappa_c(\xi, z, r(\tau, k))) \in C_P,$$
$$\tau \in [t_k^r, t_{k+1}^r], (\xi, z, r(\tau, k)) \in C_K\}$$

$$F(z, \xi, \tau, k) = (F_P(z, \kappa_c(\xi, z, r(\tau, k))),$$
$$F_K(\xi, z, r(\tau, k)), 1, 0)$$

$$D = \{x \,:\, (z, \kappa_c(\xi, z, r(\tau, k))) \in D_P,$$
$$(\tau, k) \in \mathcal{T}_r\} \cup \{x \,:\, \tau \in$$
$$[t_k^r, t_{k+1}^r), (\xi, z, r(\tau, k)) \in D_K\}$$

$$G_1(z, \xi, \tau, k) = (G_P(z, \kappa_c(\xi, z, r(\tau, k))),$$
$$\xi, \tau, k + 1),$$

$$G_2(z, \xi, \tau, k) = (z, G_K(\xi, z, r(\tau, k)), \tau, k)$$

Theorem 4 *Given a complete reference trajectory $r : \operatorname{dom} r \to \mathbb{R}^{n_P}$ and associated tracking set $\mathcal{A}$ in (9), if there exists a hybrid controller $\mathcal{H}_K$ guaranteeing that*

(1) The jumps of r and $\mathcal{H}_P$ occur simultaneously;

(2) There exist a function $V : \mathbb{R}^{n_P} \times \mathbb{R}^{n_K} \times \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}$ that is continuously differentiable; functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$; and continuous, positive definite functions ρ_1, ρ_2, ρ_3 such that

(a) For all $(z, \xi, \tau, k) \in C \cup D \cup G_1(D) \cup G_2(D)$

$$\alpha_1(|(z, \xi, \tau, k)|_\mathcal{A}) \leq V(z, \xi, \tau, k)$$
$$\leq \alpha_2(|(z, \xi, \tau, k)|_\mathcal{A})$$

(b) For all $(z, \xi, \tau, k) \in C$ and all $\zeta \in F(z, \xi, \tau, k)$,

$$\langle \nabla V(z, \xi, \tau, k), \zeta \rangle \leq -\rho_1(|(z, \xi, \tau, k)|_\mathcal{A})$$

(c) For all $(z, \xi, \tau, k) \in D_1$ and all $\zeta \in G_1(z, \xi, \tau, k)$

$$V(\zeta) - V(z, \xi, \tau, k) \leq -\rho_2(|(z, \xi, \tau, k)|_\mathcal{A})$$

(d) For all $(z, \xi, \tau, k) \in D_2$ and all $\zeta \in G_2(z, \xi, \tau, k)$

$$V(\zeta) - V(z, \xi, \tau, k) \leq -\rho_3(|(z, \xi, \tau, k)|_\mathcal{A})$$

then $\mathcal{A}$ is globally asymptotically stable.

Theorem 4 imposes that the jumps of the plant and of the reference trajectory occur simultaneously. Though restrictive, at times, this property can be enforced by proper design of the controller.

Summary and Future Directions

Advances over the last decade on modeling and robust stability of hybrid dynamical systems (without control inputs) have paved the road for the development of systematic methods for the design of control algorithms for hybrid plants. The results selected for this short expository entry, along with recent efforts on multimode/logic-based control, event-based control, and backstepping, which were not covered here, contribute to that long-term goal. The future research direction includes the development of more powerful tracking control design methods, state observers, and optimal controllers for hybrid plants.

Cross-References

▶ Lyapunov's Stability Theory
▶ Output Regulation Problems in Hybrid Systems
▶ Stability Theory for Hybrid Dynamical Systems

Bibliography

Set-Valued Dynamics and Variational Analysis:

Aubin J-P, Frankowska H (1990) Set-valued analysis. Birkhauser, Boston

Rockafellar RT, Wets RJ-B (1998) Variational analysis. Springer, Berlin/Heidelberg

Modeling and Stability:

Branicky MS (2005) Introduction to hybrid systems. In: Handbook of networked and embedded control systems. Springer, New York, pp 91–116

Haddad WM, Chellaboina V, Nersesov SG (2006) Impulsive and hybrid dynamical systems: stability,

dissipativity, and control. Princeton University Press, Princeton

Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems: modeling, stability, and robustness. Princeton University Press, Princeton

Lygeros J, Johansson KH, Simić SN, Zhang J, Sastry SS (2003) Dynamical properties of hybrid automata. IEEE Trans Autom Control 48(1):2–17

van der Schaft A, Schumacher H (2000) An introduction to hybrid dynamical systems. Lecture notes in control and information sciences. Springer, London

Control:

Biemond JJB, van de Wouw N, Heemels WPMH, Nijmeijer H (2013) Tracking control for hybrid systems with state-triggered jumps. IEEE Trans Autom Control 58(4):876–890

Forni F, Teel AR, Zaccarian L (2013) Follow the bouncing ball: global results on tracking and state estimation with impacts. IEEE Trans Autom Control 58(6):1470–1485

Lygeros J (2005) An overview of hybrid systems control. In: Handbook of networked and embedded control systems. Springer, New York, pp 519–538

Naldi R, Sanfelice RG (2013) Passivity-based control for hybrid systems with applications to mechanical systems exhibiting impacts. Automatica 49(5):1104–1116

Sanfelice RG (2013a) On the existence of control Lyapunov functions and state-feedback laws for hybrid systems. IEEE Trans Autom Control 58(12):3242–3248

Sanfelice RG (2013b) Control of hybrid dynamical systems: an overview of recent advances. In: Daafouz J, Tarbouriech S, Sigalotti M (eds) Hybrid systems with constraints. Wiley, Hoboken, pp 146–177

Sanfelice RG, Biemond JJB, van de Wouw N, Heemels WPMH (2013, to appear) An embedding approach for the design of state-feedback tracking controllers for references with jumps. Int J Robust Nonlinear Control

Hybrid Observers

Daniele Carnevale
Dipartimento di Ing. Civile ed Ing. Informatica,
Università di Roma "Tor Vergata", Roma, Italy

Abstract

In the first part of the paper, two consolidated hybrid observer designs for non-hybrid systems are presented. In the second part, recently results available in the literature related to the observability and observer design for different classes of hybrid systems are introduced.

Keywords

Introduction

Observers design, which are used to estimate the unmeasured plant state, has received a lot of attention since the late '60s. One of the first leading contribution to clearly formalize the estimation problem and propose a solution in the linear case has been proposed by Luenberger (1966). The recipe to implement a Luenberger-type observer for a continuous-time linear system described by

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \qquad (1)$$

with $x \in \mathbb{R}^n, u \in \mathbb{R}^p, y \in \mathbb{R}^m, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times p}, C \in \mathbb{R}^{m \times n}$, and $D \in \mathbb{R}^{m \times p}$, has three main ingredients: system data, the correction term commonly referred to as *output injection*, and the *observability/detectability/determinability* conditions. A Luenberger-type observer for (1), which consists in a copy of the (system data) dynamics (1) with a linear correction term $\mathbf{L}(\mathbf{y} - \hat{\mathbf{y}})$, is given by

$$\dot{\hat{x}} = A\hat{x} + Bu + \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}), \quad \hat{y} = C\hat{x} + Du, \quad (2)$$

with $L \in \mathbb{R}^{n \times m}$ and where $\hat{x}$ is the estimated value of x. The estimation error $e = x - \hat{x}$ satisfies the differential equation $\dot{e} = (A - LC)e$ with initial condition $e(0) = x(0) - \hat{x}(0)$. Since the observer has a copy of the plant dynamics and the correction term is $L(y - \hat{y}) = LCe$, the zero estimation error manifold $x = \hat{x}$ is invariant (if $x(0) = \hat{x}(0)$, then $e(t) \equiv 0$ for all $t \geq 0$), whereas its *attractivity* (yielding global exponential stability of the estimation error system) requires $A - LC$ be Hurwitz. Such an L, if A

is not already Hurwitz, exists if the pair (A, C) is *detectable* or (sufficient condition) *observable*. The observer in (2) exploits only the injection term in the for continuous time dynamics (flow map), and one may ask how profitable could be *resets* of the observer state (jump map) designing a *hybrid observer*.

The observer design for hybrid systems is a relatively new area of research and results are consolidated only for few classes of linear hybrid systems.

In section "Continuous-Time Plants," a hybrid redesign of the observer (2) is discussed first and then a more general design for nonlinear systems is introduced, whereas in section "Systems with Flows and Jumps" the recent results related to observability and observer designs for hybrid systems is discussed. Conclusions are given in section "Summary and Future Directions."

Hybrid Observers: Different Strategies

The community of researchers working on hybrid observer, which is a quite recent area and is the subject of growing interest, is wide and a unique formal definition/notation has not been reached yet. This fact is strictly related to the large number of different hybrid system models that are currently adopted by researchers. To render as simple as possible this short presentation, we let the state $x(t)$ of a hybrid system be driven by the flow map (differential equation) when $t \neq t_j$ and by the jump map (difference equation) when $t = t_j$, with $x(t)$ right continuous, i.e., $\lim_{t \to t_j^+} x(t) = x(t_j)$.

Continuous-Time Plants

Linear Case
A simple strategy to improve convergence to zero of the estimation error for (1) has been proposed in Raff and Allgower (2008) and consists in resetting the observer state x, at predetermined fixed time intervals t_j, by means of

the linear correction term $\mathbf{K(t)}\,(\mathbf{y(t)} - \mathbf{C\hat{x}(t)})$ at jump times, yielding

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + \mathbf{L}\,(\mathbf{y(t)} - \mathbf{C\hat{x}(t)}),$$
(3a)

$$\hat{x}(t_j) = x\left(t_j^-\right) + \mathbf{K\left(t_j^-\right)}\,\left(\mathbf{y\left(t_j^-\right)} - \mathbf{C\hat{x}\left(t_j^-\right)}\right),$$
(3b)

where $t_0 = 0$, $t_{j+1} - t_j = T > 0$, $j \in \mathbb{N}_{\geq 1}$ and T is a parameter that defines the interval times between resets and has to be chosen such that

$$Im\left(\lambda_p - \lambda_r\right)T \neq 2r\pi, \ r \in \mathbb{Z}\backslash\{0\}, \quad (4)$$

for each pair (λ_p, λ_r) of complex eigenvalues of the matrix $A - LC$. This preserves the (continuous time or flow) observability of the system (1) when sampled at time instants t_j and allows to select a matrix K_0 such that $(I - K_0C)\exp((A - LC)\,T)$ has all its eigenvalues at zero. Then, the estimation error $e(t)$ converges to zero in finite time (nT) if (1) is observable and the matrix $K(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n \times q}$ is selected such as $K(t) = K_0$ if $t \leq t_n$ and $K(t) = 0$ otherwise. It is important to note that the state reset (3b) yields a hybrid estimation error system given by

$$\dot{e}(t) = (A - LC)\,e(t), \quad (5a)$$

$$e(t_j) = \left(I - K\left(t_j^-\right)C\right)e\left(t_j^-\right). \quad (5b)$$

The stability property of the origin can be easily deduced by noting that

$$e(t_j) = \prod_{k=1}^{j}\left(I - K\left(t_j^-\right)C\right)$$
$$\exp\left((A - LC)\,T\right)e(0),$$

and given that $(I - K_0C)\exp((A - LC)\,T)$ is nilpotent, then $e(t_n) = e(nT) = 0$.

The potentiality benefits of hybrid observers to improve the performances of classic continuous-time observer is a relatively new area of research. Along this line, the recent work

proposed in Prieur et al. (2012) allows to limit the *peaking phenomena* for a class of *high-gain observers* opportunely resetting its (augmented) state. Moreover, when the output of (1) is a nonlinear function of the state, $y = h(x)$, with $h(\cdot)$ not invertible (e.g. the saturation function), it would be possible to rewrite (1) as a hybrid system with linear flow map and augmented state designing a hybrid observer as in Carnevale and Astolfi (2009).

Nonlinear Case

When the input of a continuous-time plant is piecewise-constant the hybrid observer proposed in Moraal and Grizzle (1995), exploiting sampled measurements, can be successfully applied for a class of nonlinear continuous (or discrete-time) systems

$$\dot{x} = f(x(t), u(t)), \, y(t) = h(x(t), u(t)), \tag{6}$$

with sufficiently smooth maps $f(\cdot, \cdot)$ and $h(\cdot, \cdot)$ and where

$$x(t_j) = F(x(t_{j-1}), u(t_{j-1})), \tag{7}$$

is the sample-data (discrete-time) version of (6) with sampling time $T = t_{j-1} - t_j$. Then, it is possible to define a hybrid observer of the following type:

$$\dot{\hat{x}}(t) = f(\hat{x}(t), u(t)), \tag{8a}$$

$$\hat{x}(t_j) = \Gamma\left(y\left(t_j^-\right), \hat{x}\left(t_j^-\right), \xi\left(t_j^-\right)\right), \tag{8b}$$

where the reset map Γ and the dynamics of the new variable $\xi(t)$ have to be properly defined. The main idea in Moraal and Grizzle (1995) is that the Newton method, in continuous and discrete time, can be used to estimate the value of ξ that renders zero the function

$$W_j^N(\xi) = Y_j^N - H\left(\xi, U_j^N\right), \tag{9}$$

where $U_j^N = \left[u'(t_{j-N+1}), \ldots, u'(t_j)\right]'$ and $Y_j^N = \left[y'(t_{j-N+1}), \ldots, y'(t_j)\right]'$ are the sampled input and output vectors, respectively, and $H : \mathbb{R}^n \times \mathbb{R}^{m \times N} \to \mathbb{R}^N$ maps the state $x(t_j)$ and the N-tuple of control inputs U_j^N into the output vector Y_j^N, i.e., $H\left(x(t_j), U_j^N\right) = Y_j^N$, and is defined as

$$H\left(x, U_j^N\right) \triangleq \begin{bmatrix} h\left(F^{-1}\left(F^{-1}(\ldots), u(t_{j-N+1})\right), u(t_{j-N+1})\right) \\ \vdots \\ h\left(F^{-1}(x, u(t_{j-1})), u(t_{j-1})\right) \\ h(x, u(t_j)) \end{bmatrix}, \tag{10}$$

where F^{-1} shortly represents the inverse of the map F such that $x(t_{j-1}) = F^{-1}(x(t_j), u(t_{j-1}))$.

The system (6)–(7) is said to be *N-osbervable,* for some $N \geq 1$ (the generic selection is $N = 2n + 1$), when $W_j^N(\xi) = 0$ hold only if $\xi = x(t_j)$, uniformly in U_j^N. Then, under certain technical assumptions (see Moraal and Grizzle 1995) related to the derivatives of f and h and the invertibility of the Jacobian matrix $J(x) = \partial H(x)/\partial x$, it is possible to select

$$\dot{\xi}(t) = kJ(\xi(t))^{-1}\left(Y_j^N \right.$$

$$\left. -H\left(\xi(t), U_j^N\right)\right), \tag{11a}$$

$$\xi(t_j) = F\left(\xi\left(t_j^-\right), u(t_{j-1})\right), \tag{11b}$$

with a sufficiently high-gain $k > 0$ and the reset map $\Gamma(\cdot) = F\left(\xi\left(t_j^-\right), u(t_{j-1})\right)$. Note that (11a) is commonly referred to as Newton flow. This approach could be easily extended to other

continuous-time minimization algorithms (normalized gradient, line-search, etc.) changing the rhs of (11a) or even with discrete-time methods iterated at higher frequency within the sample time T, yielding faster convergence to zero of the estimation error.

The same approach can be used when a continuous-time observer for (6) is considered in place of (8a) and the Newton-based resets can be used to possibly improve the performances. The continuous and discrete-time Newton algorithm require the knowledge of the jump map F to define (7), i.e. the exact discrete time model of (6), and the Jacobian matrix $J(x) = \partial H(x)/\partial x$. An approach that do not require such knowledge is proposed in Biyik and Arcak (2006), where continuous time filters and secant method allow to estimate (numerically) the map F and the Jacobian matrix, or in Sassano et al. (2011) where an *extremum-seeking*-based technique is considered.

A different approach to estimate the state of a continuous-time plant, pursued for example in Ahrens and Khalil (2009) and Liu (1997), exploits switching output injections, letting the correction term $l_\sigma(\cdot)$ to switch among opportune values selected by a suitable definition (often derived by a Lyapunov-based proof) of the switching signal $\sigma(t)$. These switching gains allow to improve observer performances and robustness against measurement noise and model uncertainties.

Systems with Flows and Jumps

The classical notion of observability does not hold for hybrid systems. As an example, consider the autonomous linear hybrid system described by $\dot{x}(t) = Ax(t)$ and $x(t_j) = Jx(t_j^-)$ with

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad J = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad (12)$$

and $C = [0, 1, 0]$. Evidently the flow is not observable in the classic sense given that $\mathcal{O}_{\text{flow}} = \begin{bmatrix} C', (CA)', (CA^2)' \end{bmatrix}'$ is not full rank and the flow-unobservable subspace

is $\ker(\mathcal{O}_{\text{flow}}) \triangleq \{x \in \mathbb{R}^3 : x_2 = x_3 = 0\}$. Nevertheless, in the first flow time interval $\tau = t_1 - t_0$, it is possible to estimate (e.g. in finite time using the observability Gramian matrix) the initial conditions $(x_2(t_0), x_3(t_0))$. Then when the first jump take place at time t_1, thanks to the structure of the jump map J that resets the value of $x_3(t_1)$ with the flow-unobservable $x_1(t_1^-)$, it is possible to estimate in the next flow time interval the value of $x_1(t_1^-)$ so that the initial condition $x(t_0)$ can be completely determined. The hybrid observability matrix in this case has the following expression

$$\mathcal{O}_{\text{hybrid}} = \begin{bmatrix} \mathcal{O}'_{flow}, (\mathcal{O}_{flow} Je^{AT_1})', \\ (\mathcal{O}_{flow}(Je^{AT_2})^2)' \end{bmatrix}'$$

and is full rank for all $T_j = t_j - t_{j-1}$ that satisfies (4). Note that from a practical point of view, in this case the time interval that allows to reconstruct the complete state is $[t_0, t_1 + \epsilon)$ since the observer needs at least an ϵ time of the new measurements (after the first jump) to evaluate the full state $\begin{bmatrix} \mathcal{O}'_{flow}, (\mathcal{O}_{flow} Je^{AT_1})', (\mathcal{O}_{flow}(Je^{AT_2})^2)' \end{bmatrix}'$. This simple example suggests that (impulsive) hybrid systems might have a reacher notion of observability than the classical ones. These properties have been studied also for mechanical systems subject to non-smooth impacts in Martinelli et al. (2004), where a high-gain-like observer design has been proposed assuming the knowledge of the impact times t_i, no *Zeno* phenomena (no finite accumulation point for t_j's), and a minimum *dwell-time,* $t_{j+1} - t_j \geq \delta > 0$. With the aforementioned assumptions and considering the more general class of hybrid system described by

$$\dot{x}(t) = f(x, u),$$
$$x(t_j) = g\left(x\left(t_j^-\right), u\left(t_j^-\right)\right), \quad (13)$$

with $y = h(x,u)$, a frequent choice is to consider the hybrid observer of the form

$$\dot{\hat{x}}(t) = f(\hat{x}, u) + \mathbf{l}(\mathbf{y}, \mathbf{x}, \mathbf{u}), \quad (14a)$$

$$\hat{x}\left(t_j\right) = g\left(\hat{x}\left(t_j^-\right), u\left(t_j^-\right)\right)$$
$$+ \mathbf{m}\left(\hat{\mathbf{x}}\left(\mathbf{t_j^-}\right), \mathbf{u}\left(\mathbf{t_j^-}\right)\right), \quad (14b)$$

with $\mathbf{l}(\cdot)$ and $\mathbf{m}(\cdot)$ that are zero when $\hat{x} = x$ rendering flow and jump-invariant the manifold $\hat{x} = x$ relying only on the correction term $l(\cdot)$ ($m \equiv 0$) in a high-gain-like design during the flow. The correction during the flow has to recover, within the minimum dwell-time δ, the worst deterioration of the estimation error induced by the jumps (if any) and the transients such that $\left\| e\left(t_{j+1}\right) \right\| < \left\| e\left(t_j^-\right) \right\|$ or $V\left(e\left(t_{j+1}\right)\right) < V\left(e\left(t_j^-\right)\right)$ if a Lyapunov analysis is considered. This type of observer design, with $m = 0$ and the linear choice $l(y, \hat{x}, u) = L(y - M\hat{x})$, have been proposed in Heemels et al. (2011) for *linear complementarity systems* (LCS) in the presence of state jumps induced by impulsive input. Therein, solutions of LCS are characterized by means of piecewise Bohl distributions and the specially defined *well-posedness* and *low-index* properties, which combined with passivity-based arguments, allow to design a global hybrid observer with exponential convergence. A separation principle to design an output feedback controller is also proposed.

An interesting approach is pursued in Forni et al. (2003) where global output tracking results on a class of linear hybrid systems subject to impacts is introduced. Therein, the key ingredient is the definition of a "mirrored" tracking reference (a change of coordinate) that depends on the sequence of different jumps between the desired trajectory (a virtual bouncing ball) and the plant (the controlled ball). Exploiting this (time-varying) change of coordinates and assuming that the impact times are known, it is possible to define an estimation error that is not discontinuous even when the tracked ball has a bounce (state jump) and the plant does not. A time regularization is included in the model embedding a minimum dwell-time among jumps. In this way, it is possible to design a linear hybrid observer represented by (14) with a linear (mirrored) term $l(\cdot)$ and $m(\cdot) \equiv 0$, proving (by standard quadratic Lyapunov functions) that the

origin of the estimation error system is GES. In this case, the standard observability condition for the couple (A, C) is required.

Switching Systems and Hybrid Automata

Switching systems and hybrid automata have been the subject of intense study of many researchers in the last two decades. For these class of systems, there is a neat separation $x = [z, q]'$ among purely discrete-time state q (*switching signal or system mode*) and rest of the state z that generically can both flow and jump. The observability of the entire system is often divided into the problem of determining the switching signal q first and then z. The switching signal can be divided into two categories: arbitrary (*universal problem*) or specific (*existential problems*) switchings.

In Vidal et al. (2003) the observability of autonomous linear switched systems with no state jump, minimum dwell time, and unknown switching signal is analyzed. Necessary and sufficient observability conditions based on rank tests and output discontinuities detection strategies are given. Along the same line, the results are extended in Babaali and Pappas (2005) to non-autonomous switched systems with non-Zeno solutions and without the minimum dwell-time requirement, providing state z and mode q observability characterized by linear-algebraic conditions.

Luenberger-type observers with two distinct gain matrices L_1 and L_2 are proposed in the case of bimodal piecewise linear systems in Juloski et al. (2007) (where state jumps are considered), whereas recently in Tanwani et al. (2013), algebraic observability conditions and observer design are proposed for switched linear systems admitting state jumps with known switching signal (although some asynchronism between the observer and the plant switches is allowed). Results related to the observability of hybrid automata, which include switching systems, can be found in Balluchi et al. (2002) and the related references. Therein the *location observer* estimates first the system *current location q*, processing system input and output assuming that it is *current-location observable*, a property that

is related to the system *current-location observation tree*. This graph is iteratively explored at each new input to determine the node associated to the current value of $q(t)$. Then, a linear (switched) Luenberger-type observer for the estimation of the state z, assuming minimum dwell-time and observability of each pair (A_q, C_q), is proposed.

Summary and Future Directions

Observer design and observability properties of general hybrid systems is an active field of research and a number of different results have been proposed although not consolidated as for classical linear systems. The results are based on different notations and definitions for hybrid systems. Efforts to provide a unified approach, in many case considering the general framework for hybrid systems proposed in Goebel et al. (2009), is pursued by the scientific community to improve consistency and cohesion of the general results. Observer designs, observability properties, and separation principle even with linear flow and jump maps are not yet completely characterized and, in the nonlinear case, only few works have been proposed (see Teel (2010)), providing open challenges for the scientific community.

Cross-References

▶ Hybrid Dynamical Systems, Feedback Control of
▶ Observer-Based Control
▶ Observers for Nonlinear Systems
▶ Observers in Linear Systems Theory

Bibliography

Ahrens JH, Khalil HK (2009) High-gain observers in the presence of measurement noise: a switched-gain approach. Automatica 45(5):936–943

Babaali M, Pappas GJ (2005) Observability of switched linear systems in continuous time. In: Morari M, Thiele L (eds) Hybrid systems: computation and control. Volume 3414 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 103–117

Balluchi A, Benvenuti L, Benedetto MDD, Vincentelli ALS (2002) Design of observers for hybrid systems. In: Hybrid systems: computation and control, Stanford, vol 2289

Biyik E, Arcak M (2006) A hybrid redesign of Newton observers in the absence of an exact discrete-time model. Syst Control Lett 55(8):429–436

Branicky MS (1998) Multiple Lyapunov functions and other analysis tools for swtiched and hybrid systems. IEEE Trans Autom Control 43(5):475–482

Carnevale D, Astolfi A (2009) Hybrid observer for global frequency estimation of saturated signals. IEEE Trans Autom Control 54(13):2461–2464

Forni F, Teel A, Zaccarian L (2003) Follow the bouncing ball: global results on tracking and state estimation with impacts. IEEE Trans Autom Control 58(8):1470–1485

Goebel R, Sanfelice R, Teel AR (2009) Hybrid dynamical systems. IEEE Control Syst Mag 29: 28–93

Heemels WPMH, Camlibel MK, Schumacher J, Brogliato B (2011) Observer-based control of linear complementarity systems. Int J Robust Nonlinear Control 21(13):1193–1218. Special issues on hybrid systems

Juloski AL, Heemels WPMH, Weiland S (2007) Observer design for a class of piecewise linear systems. Int J Robust Nonlinear Control 17(15):1387–1404

Liu Y (1997) Switching observer design for uncertain nonlinear systems. IEEE Trans Autom Control 42(12):1699–1703

Luenberger DG (1966) Observers for multivariable systems. IEEE Trans Autom Control 11: 190–197

Martinelli F, Menini L, Tornambè A (2004) Observability, reconstructibility and observer design for linear mechanical systems unobservable in absence of impacts. J Dyn Syst Meas Control 125:549

Moraal P, Grizzle J (1995) Observer design for nonlinear systems with discrete-time measurements. IEEE Trans Autom Control 40(3):395–404

Prieur C, Tarbouriech S, Zaccarian L (2012) Hybrid high-gain observers without peaking for planar nonlinear systems. In: 2012 IEEE 51st annual conference on decision and control (CDC), Maui, pp 6175–6180

Raff T, Allgower F (2008) An observer that converges in finite time due to measurement-based state updates. In: Proceedings of the 17th IFAC world congress, COEX, South Korea, vol 17, pp 2693–2695

Sassano M, Carnevale D, Astolfi A (2011) Extremum seeking-like observer for nonlinear systems. In: 18th IFAC world congress, Milano, vol 18, pp 1849–1854

Tanwani A, Shim H, Liberzon D (2013) Observability for switched linear systems: characterization and observer design. IEEE Trans Autom Control 58(5): 891–904

Teel A (2010) Observer-based hybrid feedback: a local separation principle. In: American control conference (ACC), 2010, Baltimore, pp 898–903

Vidal R, Chiuso A, Soatto S, Sastry S (2003) Observability of linear hybrid systems. In: Maler O, Pnueli A (eds) Hybrid systems: computation and control. Volume 2623 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 526–539

Identification and Control of Cell Populations

Mustafa Khammash[1] and J. Lygeros[2]
[1]Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland
[2]Automatic Control Laboratory, Swiss Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland

Abstract

We explore the problem of identification and control of living cell populations. We describe how de novo control systems can be interfaced with living cells and used to control their behavior. Using computer controlled light pulses in combination with a genetically encoded light-responsive module and a flow cytometer, we demonstrate how in silico feedback control can be configured to achieve precise and robust set point regulation of gene expression. We also outline how external control inputs can be used in experimental design to improve our understanding of the underlying biochemical processes.

Keywords

Introduction

Control systems, particularly those that employ feedback strategies, have been used successfully in engineered systems for centuries. But natural feedback circuits evolved in living organisms much earlier, as they were needed for regulating the internal milieu of the early cells. Owing to modern genetic methods, engineered feedback control systems can now be used to control in real-time biological systems, much like they control any other process. The challenges of controlling living organisms are unique. To be successful, suitable sensors must be used to measure the output of a single cell (or a sample of cells in a population), actuators are needed to affect control action at the cellular level, and a controller that connects the two should be suitably designed. As a model-based approach is needed for effective control, methods for identification of models of cellular dynamics are also needed. In this entry, we give a brief overview of the problem of identification and control of living cells. We discuss the dynamic model that can be used, as well as the practical aspects of selecting sensor and actuators. The control systems can either be realized on a computer (in silico feedback) or through genetic manipulations (in vivo feedback). As an example, we describe how de novo control systems can be interfaced with living cells and used to control their behavior. Using computer controlled light pulses in combination with a genetically encoded light-responsive module and a flow cytometer, we demonstrate how in silico feedback control can

be configured to achieve precise and robust set point regulation of gene expression.

Dynamical Models of Cell Populations

In this entry, we focus on a model of an essential biological process: gene expression. The goal is to come up with a mathematical model for gene expression that can be used for model-based control. Due to cell variability, we will work with a model that describes the average concentration of the product of gene expression (the regulated variable). This allows us to use population measurements and treat them as measurements of the regulated variable. We refer the reader to the entry ▶ Stochastic Description of Biochemical Networks in this encyclopedia for more information on stochastic models of biochemical reaction networks. In this framework, the model consist of an N-vector stochastic process $X(t)$ describing the number of molecules of each chemical species of interest in a cell. Given the chemical reactions in which these species are involve, the mean, $E[X(t)]$, of $X(t)$ evolves according to deterministic equations described by

$$\dot{E}[X(t)] = SE[w(X(t))],$$

where S is an $N \times M$ matrix that describes the stoichiometry of the M reactions described in the model, while $w(\cdot)$ is an M-vector of propensity functions. The propensity functions reflect the rate of the reactions being modeled. When one considers elementary reactions (see ▶ Stochastic Description of Biochemical Networks), the propensity function of the ith reaction, $w_i(\cdot)$, is a quadratic function of the form $w_i(x) = a_i + b_i^T x + c_i x^T Q_i x$. Typically, w_i is either a constant: $w_i(x) = a$, a linear function of the form $w_i(x) = b x_j$ or a simple quadratic of the form $w_i(x) = c x_j^2$. Following the same procedure, similar dynamical models can be derived that describe the evolution of higher-order moments (variances, covariances, third-order moments, etc.) of the stochastic process $X(t)$.

Identification of Cell Population Models

The model structure outlined above captures the fundamental information about the chemical reactions of interest. The model parameters that enter the functions $w_i(x)$ reflect the reaction rates, which are typically unknown. Moreover, these reaction rates often vary between different cells, because, for example, they depend on the local cell environment, or on unmodeled chemical species whose numbers differ from cell to cell (Swain et al. 2002). The combination of this extrinsic parameter variability with the intrinsic uncertainty of the stochastic process $X(t)$ makes the identification of the values of these parameters especially challenging.

To address this combination of intrinsic and extrinsic variabilities, one can compute the moments of the stochastic process $X(t)$ together with the cross moments of $X(t)$ and the extrinsic variability. In the process, the moments of the parametric uncertainty themselves become parameters of the extended system of ordinary differential equations and can, in principle, be identified from data. Even though doing so requires solving a challenging optimization problem, effective results can often be obtained by randomized optimization methods. For example, Zechner et al. (2012) presents the successful application of this approach to a complex model of the system regulating osmotic stress response in yeast.

When external signals are available, or when one would like to determine what species to measure when, such moment-based methods can also be used in experiment design. The aim here is to determine a priori which perturbation signals and which measurements will maximize the information on the underlying chemical process that can be extracted from experimental data, reducing the risk of conducting expensive but uninformative experiments. One can show that, given a tentative model for the biochemical process, the moments of the stochastic process $X(t)$ (and cross $X(t)$-parameter moments in the presence of extrinsic variability) can be used to approximate the Fischer information matrix and hence characterize

the information that particular experiments contain about the model parameters; an approximation of the Fischer information based on the first two moments was derived in Komorowski et al. (2011) and an improved estimate using correction terms based on moments up to order 4 was derived in Ruess et al. (2013). Once an estimate of the Fischer information matrix is available, one can design experiments to maximize the information gained about the parameters of the model. The resulting optimization problem (over an appropriate parametrization of the space of possible experiments) is again challenging but can be approached by randomized optimization methods.

Control of Cell Populations

There are two control strategies that one can implement. The control systems can be realized on a computer, using real-time measurements from the cell population to be controlled. These cells must be equipped with actuators that respond to the computer signals that close the feedback loop. We will refer to this as *in silico feedback*. Alternatively, one can implement the sensors, actuators, and control system in the entirety within the machinery of the living cells. At least in principle, this can be achieved through genetic manipulation techniques that are common in synthetic biology. We shall refer to this type of control as *in vivo feedback*. Of course some combination of the two strategies can be envisioned. In vivo feedback is generally more difficult to implement, as it involves working within the noisy uncertain environment of the cell and requires implementations that are biochemical in nature. Such controllers will work autonomously and are heritable, which could prove advantageous in some applications. Moreover, coupled with intercellular signaling mechanisms such as quorum sensing, in vivo feedback may lead to tighter regulation (e.g., reduced variance) of the cell population. On the other hand, in silico controllers are much easier to program, debug, and implement and can have much more complex dynamics that would be possible with in vivo

controllers. However, in silico controllers require a setup that maintains contact with all the cells to be controlled and cannot independently control large numbers of such cells. In this entry we focus exclusively on in silico controllers.

The Actuator

There could be several ways to send actuating signals into living cells. One consists of chemical inducers that the cells respond to either through receptors outside the cell or through translocation of the inducer molecules across the cellular membrane. The chemical signal captured by these inducers is then transduced to affect gene expression. Another approach we will describe here is induction through light. There are several light systems that can be used. One of these includes a light-sensitive protein called phytochrome B (PhyB). When red light of wavelength 650 nm is shined on PhyB in the presence of phycocyanobilin (PCB) chromophore, it is activated. In this active state it binds to another protein Pif3 with high affinity forming PhyB-Pif3 complex. If then a far-red light (730 nm) is shined, PhyB is deactivated and it dissociates from Pif3. This can be exploited for controlling gene expression as follows: PhyB is fused to a GAL4 binding domain (GAL4BD), which then binds to DNA in a specific site just upstream of the gene of interest. Pif3 in turn is fused to a GAL4 activating domain (GAL4AD). Upon red light induction, Pif3-Gal4AD complex is recruited to PhyB, where Gal4AD acts as a transcription factor to initiate gene expression. After far-red light is shined, the dissociation of GAL4BD-PhyB complex with Pif3-Gal4AD means that Gal4AD no longer activates gene expression, and the gene is off. This way, one can control gene expression – at least in open loop.

The Sensor

To measure the output protein concentration in cell populations, a florescent protein tag is needed. This tag can be fused to the protein of interest, and the fluorescence intensity emanating from each cell is a direct measure of the protein concentration in that cell. There are several technologies for measuring fluorescence of cell
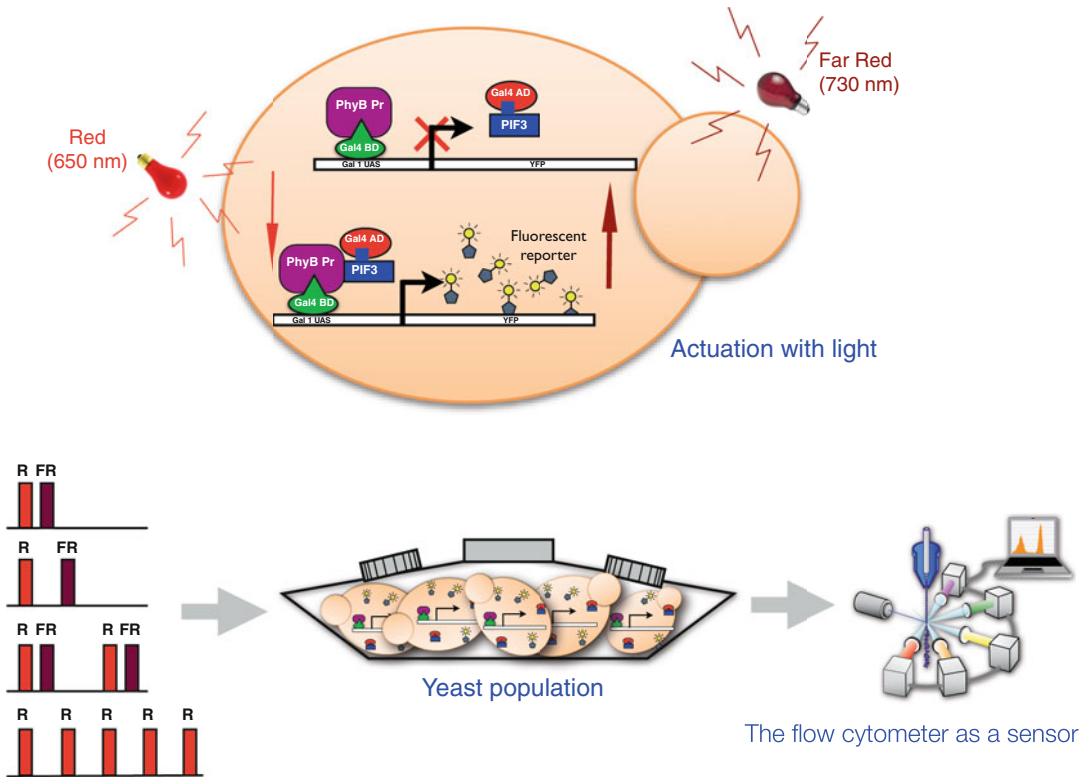
Identification and Control of Cell Populations, Fig. 1
Top figure: shows a yeast cell whose gene expression can be induced by light: red light turns on gene expression while far-red turns it off. Bottom figure: Each input light sequences can be applied to a culture of light responsive yeast cells resulting in a corresponding gene expression pattern that is measured by flow cytometry. By applying multiple carefully chosen light input test sequences and looking at their corresponding gene expression patterns a dynamic model of gene expression can be identified

populations. While fluorimeters measure the overall intensity of a population, flow cytometry and microscopy can measure the fluorescence of each individual cell in a population sample at a given time. This provides a snapshot measurement of the probability density function of the protein across the population. Repeated measurements over time can be used as a basis for model identification (Fig. 1).

The Control System

Equipped with sensors, actuators, and a model identified with the methods outlined above one can proceed to design control algorithms to regulate the behavior of living cells. Even though moment equations lead to models that look like conventional ordinary differential equations, from a control theory point of view, cell population systems offer a number of challenges. Biochemical processes, especially genetic regulation, are often very slow with time constants of the order of tens of minutes. This suggests that pure feedback control without some form of preview may be insufficient. Moreover, due to our incomplete understanding of the underlying biology, the available models are typically inaccurate, or even structurally wrong. Finally, the control signals are often unconventional; for example, for the light control system outlined above, experimental limitations imply that the system must be controlled using discrete light pulses, rather than continuous signals.

Fortunately advances in control theory allow one to effectively tackle most of these challenges. The availability of a model, for example, enables the use of model predictive control methods that introduce the necessary preview into the feedback process. The presence of unconventional inputs may make the resulting optimization problems difficult, but the slow dynamics work in our favor,
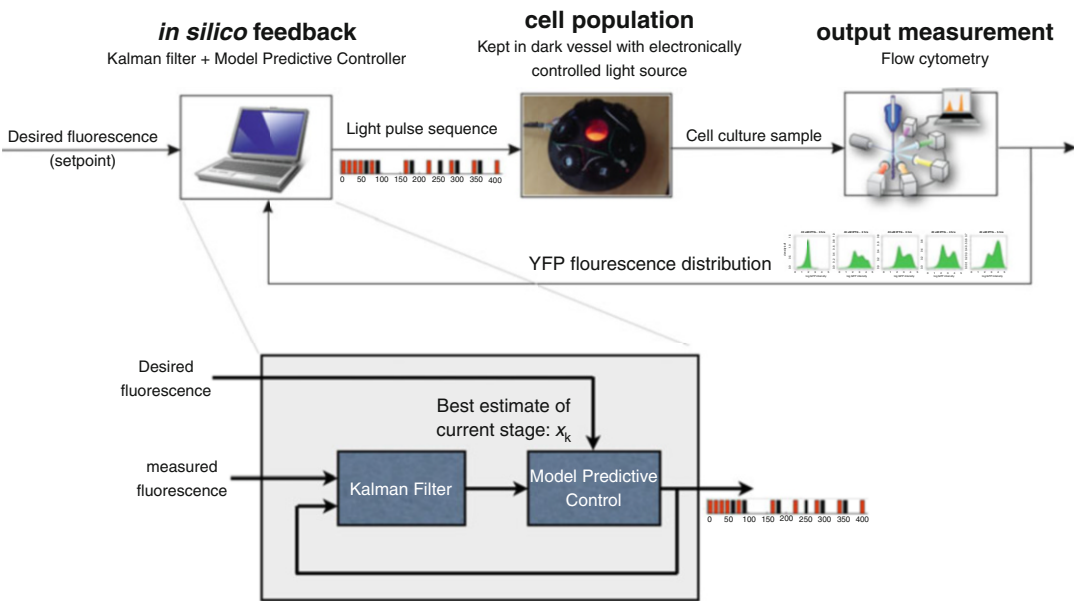
Identification and Control of Cell Populations, Fig. 2 Architecture of the closed-loop light control system. Cells are kept darkness until they are exposed to light pulse sequences from the in silico feedback controller. Cell culture samples are passed to the flow cytometer whose output is fed back to the computer which implements a Kalman filter plus a Model Predictive Controller. The objective of the control is to have the mean gene expression level follow a desired set value
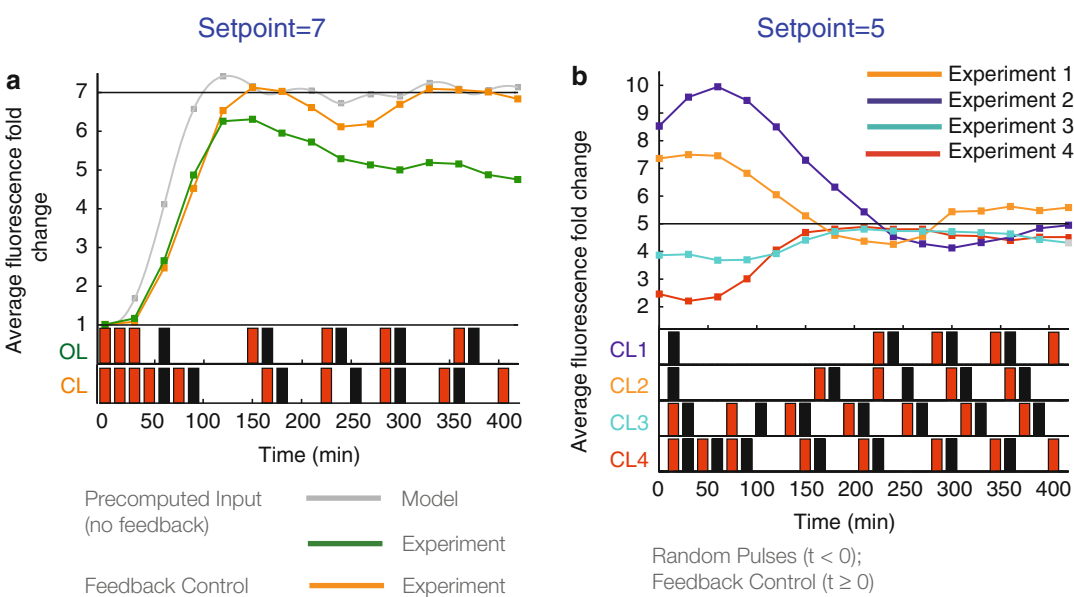
Identification and Control of Cell Populations, Fig. 3 Left panel: The closed-loop control strategy (orange) enables set point tracking, whereas an open-loop strategy (green) does not. Right panel: Four different experiments, each with a different initial condition. Closed-loop control is turned on at time t=0 shows that tracking can be achieved regardless of initial condition. (See Milias-Argeitis et al. (2011))

providing time to search the space of possible input trajectories. Finally, the fundamental principle of feedback is often enough to deal with inaccurate models. Unlike systems biology applications where the goal is to develop a model that faithfully captures the biology, in population control applications even an inaccurate model is often enough to provide adequate closed-loop performance. Exploring these issues, Milias-Argeitis et al. (2011) developed a feedback mechanism for genetic regulation using the light control system, based on an extended Kalman filter and a model predictive controller (Figs. 2 and 3). A related approach was taken in Uhlendorf et al. (2012) to regulate the osmotic stress response in yeast, while Toettcher et al. (2011) develop what is affectively a PI controller for a faster cell signaling system.

Summary and Future Directions

The control of cell populations offers novel challenges and novel vistas for control engineering as well as for systems and synthetic biology. Using external input signals and experiment design methods, one can more effectively probe biological systems to force them to reveal their secrets. Regulating cell populations in a feedback manner opens new possibilities for biotechnology applications, among them the reliable and efficient production of antibiotics and biofuels using bacteria. Beyond biology, the control of populations is bound to find further applications in the control of large-scale, multi-agent systems, including those in transportation, demand response schemes in energy systems, crowd control in emergencies, and education.

Cross-References

Bibliography

Komorowski M, Costa M, Rand D, Stumpf M (2011) Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. Proc Natl Acad Sci 108(21):8645–8650

Milias-Argeitis A, Summers S, Stewart-Ornstein J, Zuleta I, Pincus D, El-Samad H, Khammash M, Lygeros J (2011) In silico feedback for in vivo regulation of a gene expression circuit. Nat Biotech 29(12):1114–1116

Ruess J, Milias-Argeitis A, Lygeros J (2013) Designing experiments to understand the variability in biochemical reaction networks. J R Soc Interface 10:20130588

Swain P, Elowitz M, Siggia E (2002) Intrinsic and extrinsic contributions to stochasticity in gene expression. Proc Natl Acad Sci 99(20):12795–12800

Toettcher J, Gong D, Lim W, Weiner O (2011) Light-based feedback for controlling intracellular signaling dynamics. Nat Methods 8:837–839

Uhlendorf J, Miermont A, Delaveau T, Charvin G, Fages F, Bottani S, Batt G, Hersen P (2012) Long-term model predictive control of gene expression at the population and single-cell levels. Proc Natl Acad Sci 109(35):14271–14276

Zechner C, Ruess J, Krenn P, Pelet S, Peter M, Lygeros J, Koeppl H (2012) Moment-based inference predicts bimodality in transient gene expression. Proc Natl Acad Sci 109(21):8340–8345

ILC

▶ Iterative Learning Control

Industrial MPC of Continuous Processes

Mark L. Darby
CMiD Solutions, Houston, TX, USA

Abstract

Model predictive control (MPC) has become the standard for implementing constrained, multivariable control of industrial continuous processes. These are processes which are designed to operate around nominal steady-state values, which include many of the important processes found in the refining and chemical industries. The following provides an overview

of MPC, including its history, major technical developments, and how MPC is applied today in practice. Possible future developments are provided.

Keywords

Constraints; Modeling; Model predictive control; Multivariable systems; Process identification; Process testing

Introduction

Model predictive control (MPC) refer to a class of control algorithms that explicitly incorporate a process model for predicting the future response of a plant and relies on optimization as the means of determining control action. At each sample interval, MPC computes a sequence of future plant input signals that optimize future plant behavior. Only the first of the future input sequence is applied to the plant, and the optimization is repeated at subsequent sample intervals.

MPC provides an integrated solution for controlling non-square systems with complex dynamics, interacting variables, and constraints. MPC has become a standard in the continuous process industries, particularly in refining and chemicals, where it has been widely applied for over 25 years. In most commercial MPC products, an embedded steady-state optimizer is cascaded to the MPC controller. The MPC steady-state optimizer determines feasible, optimal settling values of the manipulated and controlled variables. The MPC controller then optimizes the dynamic path to optimal steady-state values.

The scope of an MPC application may include a unit operation such as a distillation column or reactor, or a larger scope such as multiple distillation columns, or a scope that combines reaction and separation sections of a plant in one controller. MPC is positioned in the control and decision hierarchy of a processing facility as shown in Fig. 1. The variables associated with MPC consist of: manipulated variables (MVs), controlled variables (CVs), and disturbance variables (DVs).
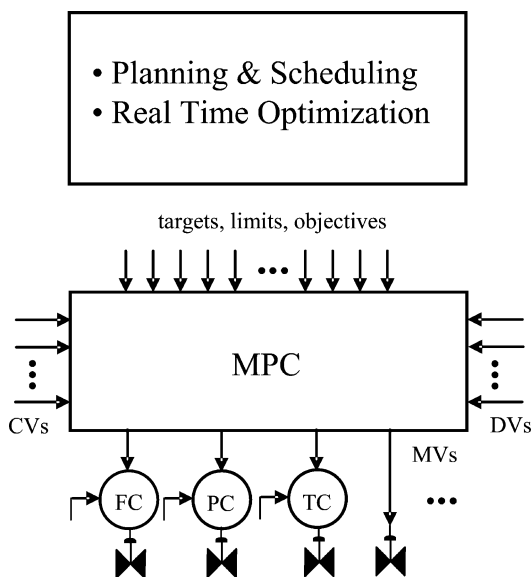
Industrial MPC of Continuous Processes, Fig. 1
Industrial control and decision hierarchy

CVs include variables normally controlled at a fixed value such as a product impurity and as well as those considered constraints, for example limits related to capacity or safety that may only be sometimes active. DVs are measurements that are treated as feedforward variables in MPC. The manipulated variables are typically setpoints of underlying PID controllers, but may also include valve position signals. Most of the targets and limits are local to the MPC, but others come directly from real-time optimization (if present), or indirectly from planning/scheduling, which are normally translated to the MPC in an open-loop manner by the operations personnel.

Linear and nonlinear model forms are found in industrial MPC applications; however, the majority of the applications continue to rely on a linear model, identified from data generated from a dedicated plant test. Nonlinearities that primarily affect system gains are often adequately controlled with linear MPC through gain scheduling or by applying linearizing static transformations. Nonlinear MPC applications tend to be reserved for those applications where nonlinearities are present in both system gains and dynamic responses and the controller must operate at significantly different targets.

Origins and History

MPC has its origins in the process industries in the 1970s. The year 1978 marked the first published description of predictive control under the name IDCOM, an acronym for Identification and Command (Richalet et al. 1978). A short time later, Cutler and Ramaker (1979) published a predictive control algorithm under the name Dynamic Matrix Control (DMC). Both approaches had been applied industrially for several years before the first publications appeared. These predictive control approaches targeted the more difficult industrial control problems that could not be adequately handled with other methods, either with conventional PID control or with advanced regulatory control (ARC) techniques that rely on single-loop controllers augmented with overrides, feedforwards/decouplers, and custom logic.

The basic idea behind the predictive control approach is shown in Fig. 2 for the case of a single input single output, stable system. Future predictions of inputs out outputs are denoted with the hat symbol and shown as dashes; double indexes, $t|k$, indicate future values at time t based on information up to and including time k. The optimization problem is to bring future predicted outputs $(\hat{y}_{k|k+1}, \ldots, \hat{y}_{k|k+P})$ close to a desired trajectory over a prediction horizon,

P, by means of a future sequence of inputs $(\hat{u}_{k|k}, \ldots, \hat{u}_{k|k+M-1})$ calculated over a control horizon M. The trajectory may be a constant setpoint. In the general case, the optimization is performed subject to constraints that may be imposed on future inputs and outputs. Only the first of the future moves is implemented and the optimization is repeated at the next time instant. Feedback, which accounts for unmeasured disturbances *and* model error, is incorporated by shifting all future output predictions, prior to the optimization, based on the difference between the output measurement y_k and the previous prediction $\hat{y}_{k|k-1}$, denoted by $d_{k|k}$ (i.e., the prediction error at time instant k). Future predicted values of the outputs depend on both past *and* future values inputs. If no future input changes are made (at time k or after), the model can be used to calculate the future "free" output response, $y_{t:k}^0$, which will ultimately settle at a new steady-state value based on the settling time (or time to steady state of the model, T_{ss}). For the unconstrained case, it is straightforward to show that the optimal result is a linear control law that depends only on the error between the desired trajectory and the free output response.

The predictive approach seemed to contrast with the state-space optimal control method of the time, the linear quadratic regulator (LQR). Later research exposed the similarities to LQR

Industrial MPC of Continuous Processes, Fig. 2 Predictive control approach
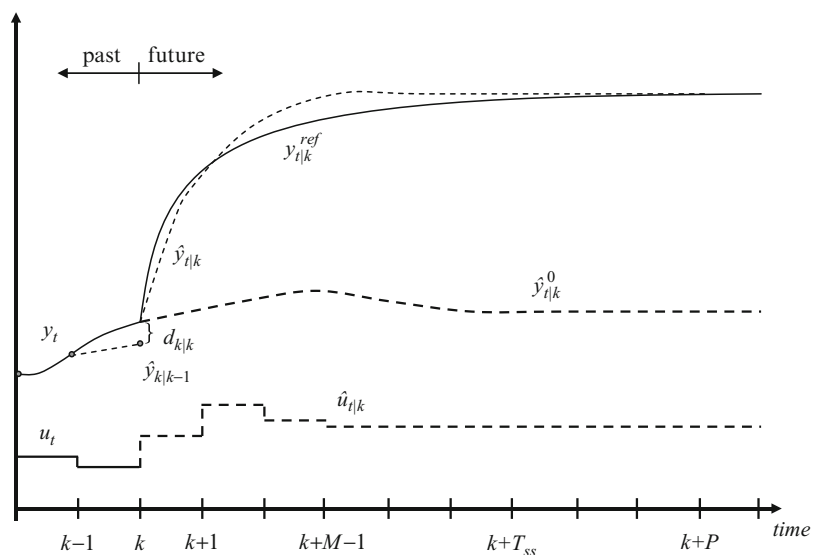
and also Internal Model Control (IMC) (Garcia and Morari 1982), although these techniques did not solve an online optimization problem. Optimization-based control approaches became feasible for industrial applications due to (1) the slower sampling requirements of most industrial control problems (on the order of minutes) and the hierarchical implementations in which MPC provides setpoints to lower level PID controllers which execute on a much faster sample time (on the order of seconds or faster).

Although the basic ideas behind MPC remain, industrial MPC technology has changed considerably since the first formulations in the late 1970s. Qin and Badgwell (2003) describe the enhancements to MPC technology that occurred over the next 20 plus years until the late 1990s. Enhancements since that time are highlighted in Darby and Nikolaou (2012). These improvements to MPC reflect increases in computer processing capability and additional requirements of industry, which have led to increased functionality and tools/techniques to simplify implementation. A summary of the significant enhancements that have been made to industrial MPC is highlighted below.

Constraints: Posing input and output constraints as linear inequalities, expressed as a function of the future input sequence (Garcia and Morshedi 1986), and solved by a standard quadratic program or an iterative scheme which approximates one.

Two-Stage Formulations: Limitations of a single objective function led to two-stage formulations to handle MV degrees of freedom (constraint pushing) and steady-state optimization via a linear program (LP).

Integrators. In their native form, impulse and step response models can be applied only to stable systems (in which the impulse response model coefficients approach zero). Extension to handle integrating variables included embedding a model of the difference of the integrating signal or integrating a fraction of the current prediction error into the future (implying an increasing $\left|d_{k+j|k}\right|$ for $j \geq 1$ in Fig. 2). The desired value of an integrator at steady state (e.g., zero slope) has been incorporated into two-stage formulations (see, e.g., Lee and Xiao 2000).

State Space Models. The first state space formulation of MPC, which was introduced in the late 1980s (Marquis and Broustail 1988) allowed MPC to be extended to integrating *and* unstable processes. It also made use of the Kalman filter which provided additional capability to estimate plant states and unmeasured disturbances. Later, a state space MPC offering was developed based on an infinite horizon (for both control and prediction) (Froisy 2006). These state space approaches provided a connection back to unconstrained LQR theory.

Nonlinear MPC. The first applications of nonlinear MPC, which appeared in the 1990s, were based on neural net models. In these approaches, a linear dynamic model was combined with a neural net model that accounted for static nonlinearity (Demoro et al. 1997; Zhao et al. 2001).

The late 1990s saw the introduction of an industrial nonlinear MPC based on first principle models derived from differential mass and energy balances and reaction kinetic expressions, expressed in differential algebraic equation (DAE) form (Young et al. 2002).

A process where nonlinear MPC is routinely applied is polymer manufacturing.

Identification Techniques. Multivariable prediction error techniques are now routinely used. More recently, industrial application of subspace identification methods has appeared, following the development of these algorithms in the 1990s. Subspace methods incorporate the correlation of output measurements in the identification of a multivariable state space model, which can be used directly in a state space MPC or converted to an impulse or step response model based MPC.

Testing Methods. The 1990s saw increased use of automatic testing methods to generate data for (linear) dynamic model identification using uncorrelated binary signals. Since the 2000, closed-loop testing methods have received considerable attention.

The motivation for closed-loop testing is to reduce implementation time and/or effort of the initial implementation as well as the ongoing need to re-identify the model of an industrial application in light of processes changes. These closed-loop testing methods, which require a preliminary or existing model, utilize uncorrelated dither signals either introduced as biases to the controller MVs or injected through the steady-state LP or QP, where additional logic or optimization of the test protocol may be performed (Kalafatis et al. 2006; MacArthur and Zhan 2007; Zhu et al. 2012).

Mathematical Formulation

While there are differences in how the MPC problem is formulated and solved, the following general form captures most of the MPC products (Qin and Badgwell 2003), although not all terms may be present in a given product:

$$
\min_{\Delta \mathcal{U}}
\begin{bmatrix}
\sum_{j=1}^{P} \left\| \hat{\mathbf{y}}_{k+j|k} - \mathbf{y}_{k+j|k}^{ref} \right\|_{\mathbf{Q}_j}^2 + \sum_{j=1}^{P} \left\| \mathbf{s}_{k+j|k} \right\|_{\mathbf{T}_j}^2 \\
\sum_{j=1}^{M-1} \left\| \hat{\mathbf{u}}_{k+j|k} - \mathbf{u}^{ss} \right\|_{\mathbf{R}_j}^2 + \sum_{j=1}^{M-1} \left\| \Delta \mathbf{u}_{k+j|k} \right\|_{\mathbf{S}_j}^2
\end{bmatrix}
\tag{1}
$$

subject to:

$$
\left.
\begin{aligned}
\hat{\mathbf{x}}_{k+j|k} &= \mathbf{f}(\hat{\mathbf{x}}_{k+j-1|k}, \hat{\mathbf{u}}_{k+j-1|k}), \quad j = 1, \dots P \\
\hat{\mathbf{y}}_{k+j|k} &= \mathbf{g}(\hat{\mathbf{x}}_{k+j|k}, \hat{\mathbf{u}}_{k+j|k}), \quad j = 1, \dots P
\end{aligned}
\right\} \text{ Model equations}
$$

$$
\left.
\begin{aligned}
\mathbf{y}^{\min} - \mathbf{s}_j &\le \hat{\mathbf{y}}_{k+j|k} \le \mathbf{y}^{\max} + \mathbf{s}_j, \quad j = 1, \dots, P \\
\mathbf{s}_j &\ge 0, \quad j = 1, \dots P
\end{aligned}
\right\} \text{ Output constraints/slacks}
$$

$$
\left.
\begin{aligned}
\mathbf{u}^{\min} &\le \hat{\mathbf{u}}_{k+j|k} \le \mathbf{u}^{\max}, \quad j = 0, \dots M-1 \\
-\Delta \mathbf{u}^{\min} &\le \Delta \mathbf{u}_{k+j|k} \le \Delta \mathbf{u}^{\max}, \quad j = 0, \dots M-1
\end{aligned}
\right\} \text{ Input constraints}
$$

where the minimization is performed over the future sequence of inputs $\mathcal{U} \triangleq \hat{\mathbf{u}}_{k|k}, \hat{\mathbf{u}}_{k+1|k}, \dots, \hat{\mathbf{u}}_{k+M-1|k}$. The four terms in the objective function represent conflicting quadratic penalties ($\|\mathbf{x}\|_{\mathbf{A}}^2 \triangleq \mathbf{x}^T \mathbf{A}\mathbf{x}$); the penalty matrices are most always diagonal. The first term penalizes the error relative to a desired reference trajectory (cf. Fig. 2) originating at $\hat{\mathbf{y}}_{k|k}$ and terminating at a desired steady-state, $\mathbf{y}^{ss}$; the second term penalizes output constraint violations over the prediction horizon (constraint softening); the third term penalizes inputs deviations from a desired steady-state, either manually specified or calculated. The fourth term penalizes input changes as a means of trading off output tracking and input movement (move suppression).

The above formulation applies to both linear and nonlinear MPC. For linear MPCs, except for state space formulations, there are no state

equations and the outputs in the dynamic model are a function of only past inputs, such as with the finite step response model.

When a steady-state optimizer is present in the MPC, it provides the steady-state targets for $\mathbf{u}^{ss}$ (in the third quadratic term) and $\mathbf{y}^{ss}$ (in the output reference trajectory). Consider the case of linear MPC with LP as the steady-state optimizer. The LP is typically formulated as

$$
\min_{\Delta \mathbf{u}^{ss}} \mathbf{c}_u^T \Delta \mathbf{u}^{ss} + \mathbf{c}_y^T \Delta \mathbf{y}^{ss} + \mathbf{q}_+^T \mathbf{s}^+ + \mathbf{q}_-^T \mathbf{s}^-
$$

subject to:

$$
\left.
\begin{aligned}
\Delta \mathbf{y}^{ss} &= \mathbf{G}^{ss} \Delta \mathbf{u}^{ss} \\
\mathbf{u}^{ss} &= \mathbf{u}_{k-1} + \Delta \mathbf{u}^{ss} \\
\mathbf{y}^{ss} &= \mathbf{y}_{k+T_{ss}|k}^0 + \Delta \mathbf{y}^{ss}
\end{aligned}
\right\} \text{ Model equations}
$$

$$\left. \mathbf{y}^{\min} - \mathbf{s}^- \le \mathbf{y}^{ss} \le \mathbf{y}^{\max} + \mathbf{s}^+ \right\} \text{ Output constraints}$$

$$\left. \begin{array}{c} \mathbf{u}^{\min} \le \mathbf{u}^{ss} \le \mathbf{u}^{\max} \\[6pt] -M\Delta\mathbf{u}^{\max} \le \mathbf{u}^{ss} \le M\Delta\mathbf{u}^{\max} \end{array} \right\} \text{ Input constraints}$$

$\mathbf{G}^{ss}$ is formed from the gains of the linear dynamic model. Deviations outside minimum and maximum output limits ($\mathbf{s}^-$ and $\mathbf{s}^+$, respectively) are penalized, which provide constraint softening in the event all outputs cannot be simultaneously controlled within limits. The weighting in $\mathbf{q}_-$ and $\mathbf{q}_+$ determine the relative priorities of the output constraints. The input constraints, expressed in terms of $\Delta\mathbf{u}^{\max}$, prevent targets from being passed to the dynamic optimization that cannot be achieved. The resulting solution – $\mathbf{u}^{ss}$ and $\mathbf{y}^{ss}$ – provides a consistent, achievable steady-state for the dynamic MPC controller. Notice that for inputs, the steady-state delta is applied to the current value and, for outputs, the steady-state delta is applied to the steady-state prediction of the output without future moves, after correcting for the current model error (cf. Fig. 2). If a real-time optimizer is present, its outputs, which may be targets for CVs and/or MVs, are passed to the MPC steady-state optimizer and considered with other objectives but at lower weights or priorities.

Some additional differences or features found in industrial MPCs include:

1-Norm formulations where absolute deviations, instead of quadratic deviations, are penalized.

Use of zone trajectories or "funnels" with small or no penalty applied if predictions remain within the specified zone boundaries.

Use of a minimum movement criterion in either the dynamic or steady-state optimizations, which only lead to MV movement when CV predictions go outside specified limits. This can provide controller robustness to modeling errors.

Multiobjective formulations which solve a series of QP or LP problems instead of a single one, and can be applied to the dynamic or steady-state optimizations. In these formulations, higher priority objectives are solved first, followed by lesser priority objectives with the solution of the higher priority objectives

becoming equality constraints in subsequent optimizations (Maciejowski 2002).

MPC Design

Key design decision for a given application are the number of MPC controllers and the selection of the MVs, DVs, and CVs for each controller; however, design decisions are not limited to just the MPC layer. The design problem is one of deciding on the best overall structure for the MPC(s) *and* the regulatory controls, given the control objectives, expected constraints, qualitative knowledge of the expected disturbances, and robustness considerations. It may be that existing measurements are insufficient and additional sensors may be required. In addition, a measurement many not be updated on a time interval consistent with acceptable dynamic control, for example, laboratory measurements and process composition analyzers. In this case, a soft sensor, or inferential estimator, may need to be developed from temperature and pressure measurements.

MPC is frequently applied to a major plant unit, with the MVs selected based on their sensitivity to key unit CVs and plant economics. Decisions regarding the number and size of the MPCs for a given application depend on plant objectives, (expected) constraints, and also designer preferences. When the objective is to minimize energy consumption based on fixed or specified feed rate, multiple smaller controllers can be used. In this situation, controllers are normally designed based on the grouping of MVs with the largest effect on the identified CVs, often leading to MPCs designed for individual sections of equipment, such as reactors and distillation columns. When the objective is to maximize feed (or certain products), larger controllers are normally designed, especially if there are multiple constraints that can limit plant throughput. The MPC steady-state LP or QP is ideally suited to solving the throughput maximization problem by utilizing all available MVs. The location of the most limiting constraints can impact the number of MPCs. If the major constraints are near the front-end of the plant, one MPC can be designed

which connects these constraints with key MVs such as feed rates, and other MPCs designed for the rest of the plant. If the major constraints are located near the back of the plant, then a single MPC is normally considered; alternatively, an MPC cascade could be considered, although this is not a common practice across the industry (and often requires customization).

The feed maximization objective is a major reason why MPCs have become larger with the advances in computer processing capability. However, there is generally a higher requirement on model consistency for larger controllers due do the increased number of possible constraint sets against which the MPC can operate. A larger controller can also be harder to implement and understand. This is a reason why some practitioners prefer implementing smaller MPCs at the potential loss of benefits.

MPC Practice

An MPC project is typically implemented in the following sequence:
Pretest and preliminary MPC design
Plant testing
Model and controller development
Commissioning
These tasks apply whether the MPC is linear or nonlinear, but with some differences, primarily model development and in plant testing. In nonlinear MPC, key decisions are related to the model form and level of rigor. Note that with a fundamental model, lower level PID loops must be included in the model, if the dynamics are significant; this is in contrast to empirical modeling, where the dynamics of the PID loops are embedded in the plant responses. A fundamental model will typically require less plant testing and make use of historical operating data to fit certain model parameters such as heat transfer coefficients and reaction constants. Historical data and/or data from a validated nonlinear static model can also be used to develop nonlinear static models (e.g., neural net) to combine with empirical dynamic models. As mentioned earlier, most industrial applications continue to rely on

empirical linear dynamic models, fit to data from a dedicated plant test. This will be the basis in the following discussion.

In the pretest phase of work, the key activity is one of determining the base level regulatory controls for MPC, tuning of these controls, and determining if the current plant instrumentation is adequate. It is common to retune a significant number of PID loops, with significant benefits often resulting from this step alone.

A range of testing approaches are used in plant testing for linear MPC, including both manual and automatic (computer-generated) test signal designs, most often in open loop but, increasingly, in closed loop. Most input testing continues to be based on uncorrelated signals, implemented either manually or from computer-generated random sequences. Model accuracy requirements dictate accuracy across a range of frequencies which is achieved by varying the duration of the steps. Model identification runs are made through out the course of a test to determine when model accuracy is sufficient and a test can be stopped.

In the next phase of work, modeling of the plant is performed. This includes constructing the overall MPC model from individual identification runs; for example, deciding which models are significant and judging the models characteristics (dead times, inverse response settling time, gains) based on engineering/process and a priori knowledge. An important step is analyzing, and adjusting if necessary, the gains of the constructed model to insure the models gains satisfy mass balances and gain ratios do not result in fictitious degrees of freedom (due to model errors) that the steady-state optimizer could exploit. Also included is the development of any required inferentials or soft sensors, typically based on multivariate regression techniques such as principal component regression (PCR), principal component analysis (PCA) and partial least squares (PLS), or sometimes based on a fundamental model.

During controller development, initial controller tuning is performed. This relates to establishing criteria for utilizing available degrees of freedom and setting control variable priorities. In addition, initial tuning values are established for

the dynamic control. Steady state responses corresponding to expected constraint scenarios are analyzed to determine if the controller behaves as expected, especially with respect to the steady-state changes in the manipulated variables.

Commissioning involves testing and tuning the controller against different constraint sets. It is not unusual to modify or revisit model decisions made earlier. In the worst case, control performance may be deemed unacceptable and the control engineer is forced to revisit earlier decisions such as the base level regulatory strategy or plant model quality, which would require re-testing and re-identification of portions of the plant model. The main commissioning effort typically takes place over a two to three week period, but can vary based on the size and model density of the controller. In reality, commissioning, or more accurately, controller maintenance, is an ongoing activity. It is important that the operating company have in-house expertise that can be used to answer questions ("why is the controller doing that?"), troubleshoot, and modify the controller to reflect new operating modes and constraint sets.

Future Directions

Likely future developments are expected to follow extensions of current approaches. Due to the success in automatic, closed-loop testing, one possibility is extending it to "dual" or "joint" control, where control and identification objectives are combined and allow the user to select how much the control (e.g., output variance) can be affected by test perturbation signals. Another is in formulating the plant test as a DOE 8 (design of experiments) optimization problem that could, for example, target specific models or model parameters. In the identification area, extensions have started to appear which allow constraints to be imposed, for example, on dead-times or gains, thus allowing a priori knowledge to be used. Another important area that has seen recent emphasis, and which more development can be expected, is in monitoring and diagnosis, for example, detecting which submodels of MPC have become inaccurate and require re-identification.

As mentioned earlier, one of the advantages of state-space modeling is the inherent flexibility to model unmeasured disturbances (i.e., $d_{k+j|j}$, cf. Fig. 2); however, these have not found widespread use in industry. A useful enhancement would be a framework for developing and implementing improved estimators in a convenient and transparent manner, that would be applicable to traditional FIR- and FSR- based MPCs.

In the area of nonlinear control, the use of hybrid modeling approaches has increased, for example, integrating known fundamental model relationships with neural net or linear time-varying dynamic models. The motivation is in reducing complexity and controller execution times. The use of hybrid techniques can be expected to further increase, especially if nonlinear control is to be applied more broadly to larger control problems. Even in situations where control with linear MPC is adequate, there may be benefits from the use of hybrid or fundamental models, even if the models are not directly used in the control calculation. The resulting model could be used offline in model development or online to update the linear MPC model. Benefits would come from reduced plant testing and in ensuring model consistency. In the longer term, one can foresee a more general modeling and control environment where the user would not have to be concerned with the distinction between linear and nonlinear models and would be able to easily incorporate known relationships into the controller model.

An area that has not received significant attention, but is suggested as an area worth pursuing concerns MPC cascades. Most of the applications and research are based on a single MPC or multiply distributed MPCs. An MPC cascade would permit the lower MPC to run at a faster time period and allow the user to decide which degrees of freedom are to be used for higher level objectives, such as feed maximization.

Cross-References

▶ Control Hierarchy of Large Processing Plants: An Overview
▶ Control Structure Selection

▶ Model-Based Performance Optimizing
  Control
▶ Model-Predictive Control in Practice
▶ Nominal Model-Predictive Control
▶ Real-Time Optimization of Industrial Processes

Bibliography

Cutler CR, Ramaker BL (1979) Dynamic matrix control –
  a computer control algorithm. In: AIChE national
  meeting, Houston
Darby ML, Nikolaou M (2012) MPC: current practice and
  challenges. Control Eng Pract 20(4):328–352
Demoro E, Axelrud C, Johnson D, Martin G (1997) Neural network modeling and control of polypropylene
  process. In: Society of plastic engineers international
  conference, Houston, 487–799
Froisy JB (2006) Model predictive control – building a
  bridge between theory and practice. Comput Chem
  Eng 30:1426–1435
Garcia CE, Morari M (1982) Internal model control. 1. A
  unifying review and some new results. Ind Eng Chem
  Process Des Dev 21:308–323
Garcia CE, Morshedi AM (1986) Quadratic programming
  solution of dynamic matrix control (QDMC). Chem
  Eng Commun 46(1–3):73–87
Kalafatis K, Patel K, Harmse M, Zheng Q, Craig M (2006)
  Multivariable step testing for MPC projects reduces
  crude unit testing time. Hydrocarb Process 86:93–99
Lee JH, Xiao J (2000) Use of two-stage optimization
  in model predictive control of stable and integrating.
  Comput Chem Eng 24:1591–1596
MacArthur JW, Zhan C (2007) A practical multi-stage
  method for fully automated closed-loop identification
  of industrial processes. J Process Control 17(10):770–
  786
Maciejowski J (2002) Predictive control with constraints.
  Prentice Hall, Englewood Cliffs
Marquis P, Broustail JP (1988) SMOC, a bridge between
  state space and model predictive controllers: application to the automation of a hydrotreating unit. In:
  Proceedings of the 1988 IFAC workshop on model
  based process control, Atlanta, GA, 37–43
Qin JQ, Badgwell TA (2003) A survey of industrial model
  predictive control. Control Eng Pract 11:733–764
Richalet J, Rault A, Testud J, Papon J (1978) Model
  predictive control: applications to industrial processes.
  Automatica 14:413–428
Young RE, Bartusiak D, Fontaine RW (2002) Evolution
  of an industrial nonlinear model predictive controller.
  In: Sixth international conference on chemical process control, Tucson. CACHE, American Institute of
  Chemical Engineers, 342–351
Zhao H, Guiver JP, Neelakantan R, Biegler L (2001) A
  nonlinear industrial model predictive controller using
  integrated PLS and neural net state space model. Control Eng Pract 9(2): 125–133
Zhu Y, Patwardhan R, Wagner S, Zhao J (2012) Towards
  a low cost and high performance MPC: the role of
  system identification. Comput Chem Eng 51(5):124–
  135

Information and Communication Complexity of Networked Control Systems

Serdar Yüksel
Department of Mathematics and Statistics,
Queen's University, Kingston, ON, Canada

Abstract

Information and communication complexity of
a networked control system identifies the minimum amount of information exchange needed
between the decision makers (such as encoders,
controllers, and actuators) to achieve a certain
objective, which may be in terms of reaching a
target state or achieving a given cost threshold.
This formulation does not impose any constraints
on the computational requirements to perform the
communication or control. Both stochastic and
deterministic formulations are considered.

Keywords

Communication complexity; Information theory;
Networked control

Introduction

Consider a dynamic team problem with L control
stations (these will be referred to as decision
makers and denoted by DMs) under the following
dynamics and measurement equations:

$$x_{t+1} = f_t(x_t, u_t^1, \ldots, u_t^L, w_t), \qquad t = 0, 1, \cdots \tag{1}$$

$$y_t^i = g_t^i(x_t, u_{t-1}^1, \ldots, u_{t-1}^L; v_t^i), \tag{2}$$

**Information and
Communication
Complexity of
Networked Control
Systems, Fig. 1** A
decentralized networked
control system with
information exchange
between decision makers
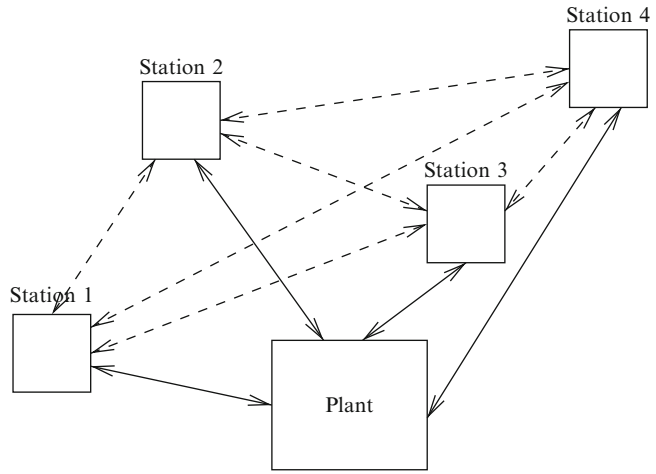
where $i \in \{1, 2, \ldots, L\} =: \mathcal{L}$ and $x_0, w_{[0,T-1]}$, $v_{[0,T-1]}^i$ are mutually independent random variables with specified probability distributions. Here, we use the notation $w_{[0,t]} := \{w_s, 0 \leq s \leq t\}$.

The DMs are allowed to exchange limited information: see Fig. 1. The information exchange is facilitated by an encoding protocol $\mathcal{E}$ which is a collection of admissible encoding functions described as follows. Let the information available to DM i at time t be

$$\mathcal{I}_t^i = \{y_{[1,t]}^i, u_{[1,t-1]}^i, z_{[0,t]}^{i,j}, z_{[0,t]}^{j,i}, j \in \mathcal{L}\},$$

where $z_t^{i,j}$ takes values in $\mathcal{Z}_t^{i,j}$ and is the information variable transmitted from DM i to DM j at time t generated with

$$\mathbf{z}_t^i = \{z_t^{i,j}, j \in \mathcal{L}\} = \mathcal{E}_t^i(\mathcal{I}_{t-1}^i, u_{t-1}^i, y_t^i), \quad (3)$$

and for $t = 0$, $\mathbf{z}_0^i = \{z_0^{i,j}, j \in \mathcal{L}\} = \mathcal{E}_0^i(y_0^i)$. The control actions are generated with

$$u_t^i = \gamma_t^i(\mathcal{I}_t^i),$$

for all DMs. Define $\log_2(|\mathcal{Z}_t^{i,j}|)$ to be the *communication rate from DM i to DM j at time t* and $\mathcal{R}(\mathbf{z}_{[0,T-1]}) = \sum_{t=0}^{T-1} \sum_{i,j \in \mathcal{L}} \log_2(|\mathcal{Z}_t^{i,j}|)$ to be the *(total) communication rate*. The minimum (total) communication rate over all coding and control policies subject to a design objective

is called the *communication complexity* for this objective.

The above is a fixed-rate formulation for communication complexity, since for any two coder outputs, a fixed number of bits is used at any given time. One could also use variable-rate formulations. The variable-rate formulation exploits the probabilistic distribution of the system variables: see Cover and Thomas (1991).

Communication Complexity for Decentralized Dynamic Optimization

Let $\underline{\mathcal{E}}^i = \{\mathcal{E}_t^i, t \geq 0\}$ and $\underline{\gamma}^i = \{\gamma_t^i, t \geq 0\}$. Under a team-encoding policy $\underline{\mathcal{E}} = \{\underline{\mathcal{E}}^1, \underline{\mathcal{E}}^2, \ldots, \underline{\mathcal{E}}^L\}$, and a team-control policy $\underline{\gamma} = \{\underline{\gamma}^1, \underline{\gamma}^2, \ldots, \underline{\gamma}^L\}$, let the induced cost be

$$E^{\underline{\gamma}, \underline{\mathcal{E}}}[\sum_{t=0}^{T-1} c(x_t, u_t^1, u_t^2, \cdots, u_t^L)]. \quad (4)$$

In networked control, the goal is to minimize (4) over all coding and control policies subject to information constraints in the system. Let $\mathbf{u}_t = \{u_t^1, u_t^2, \cdots, u_t^L\}$. The following definition and example are from Yüksel and Başar (2013).

Definition 1 Given a decentralized control problem as above, *team cost-rate function* $C : \mathbb{R} \to \mathbb{R}$ is

$$C(R) := \inf_{\underline{\gamma}, \underline{\mathcal{E}}} \left\{ E^{\underline{\gamma}, \underline{\mathcal{E}}}[\sum_{t=0}^{T-1} c(x_t, \mathbf{u}_t)] : \right.$$

$$\left. \frac{1}{T}\mathcal{R}(\mathbf{z}_{[0,T-1]}) \leq R \right\}.$$

We can define a dual function.

Definition 2 Given a decentralized control problem as above, *team rate-cost function* $R : \mathbb{R} \to \mathbb{R}$ is

$$R(C) := \inf_{\underline{\gamma}, \underline{\mathcal{E}}} \left\{ \frac{1}{T}\mathcal{R}(\mathbf{z}_{[0,T-1]}) : \right.$$

$$\left. E^{\underline{\gamma}, \underline{\mathcal{E}}}[\sum_{t=0}^{T-1} c(x_t, \mathbf{u}_t)] \leq C \right\}.$$

The formulation here can be adjusted to include sequential (iterative) information exchange given a fixed ordering of actions, as opposed to a simultaneous (parallel) information exchange at any given time t. That is, instead of (3), we may have

$$\mathbf{z}_t^i = \{z_t^{i,j}, j \in \{1, 2, \ldots, L\}\}$$
$$= \mathcal{E}_t^i(\mathcal{I}_{t-1}^i, u_{t-1}^i, y_t^i, \{z_t^{k,i}, k < i\}). \quad (5)$$

Both to make the discussion more explicit and to show that a sequential (iterative) communication protocol may perform strictly better than an optimal parallel communication protocol given a total rate constraint, we state the following example: Consider the following setup with two DMs. Let x^1, x^2, p be uniformly distributed binary random variables, DM i have access to y^i, $i = 1, 2$, and

$$x = (p, x^1, x^2), \qquad y^1 = p, \qquad y^2 = (x^1, x^2),$$

and the cost function be

$$c(x, u^1, u^2) = 1_{\{p=0\}}c(x^1, u^1, u^2)$$
$$+ 1_{\{p=1\}}c(x^2, u^1, u^2),$$

with

$$c(s, u^1, u^2) = (s - u^1)^2 + (s - u^2)^2.$$

Suppose that we wish to compute the minimum expected cost subject to a total rate of 2 bits that can be exchanged. Under a sequential scheme, if we allow DM 1 to encode y^1 to DM 2 with 1 bit, then a cost of 0 is achieved since DM 2 knows the relevant information that needs to be transmitted to DM 1, again with 1 bit: If $p = 0$, x^1 is the relevant random variable with an optimal policy $u^1 = u^2 = x^1$, and if $p = 1$, x^2 is relevant with an optimal policy $u^1 = u^2 = x^2$, and a cost of 0 is achieved. However, if the information exchange is parallel, then DM 2 does not know which state is the relevant one, and it can be shown that a cost of 0 cannot be achieved under any policy.

The formulation in Definition 1 can also be adjusted to allow for multiple rounds of communication per time stage. Having multiple rounds can enhance the performance for a class of team problems while keeping the total rate constant.

Communication Complexity in Decentralized Computation

Yao (1979) initiated the research on communication complexity in distributed computation. This may be viewed as a special case of the setting considered earlier but with finite spaces and in a deterministic and an error-free context: Consider two decision makers (DMs) who have access to local variables $x \in \{0,1\}^n$, $y \in \{0,1\}^n$. Given a function f of variables (x, y), what is the maximum (over all input variables x, y) of the minimum amount of information exchange needed for at least one agent to compute the value of the function? Let $s(x, y) = \{m_1, m_2, \cdots, m_t\}$ be the communication symbols exchanged on input (x, y) during the execution of a communication protocol. Let m_i denote the ith binary message symbol with $|m_i|$ bits. The communication complexity for such a setup is defined as

$$R(f) = \min_{\underline{\gamma}, \underline{\mathcal{E}}} \max_{(x,y) \in \{0,1\}^n \times \{0,1\}^n} |s(x, y)|, \quad (6)$$

where $|s(x, y)| = \sum_{i=1}^{t} |m_i|$ and $\underline{\mathcal{E}}$ is a protocol which dictates the iterative encoding functions as in (5) and $\underline{\gamma}$ is a decision policy.

For such problems, obtaining *good* lower bounds is in general challenging. One lower bound for such problems is obtained through the following reasoning: A subset of the form $A \times B$, where A and B are subsets of $\{0, 1\}^n$, is called an f-*monochromatic* rectangle if for every $x \in A, y \in B$, $f(x, y)$ is the same. It can be shown that given any finite message sequence $\{m_1, m_2, \cdots, m_t\}$, the set $\{(x, y) : s(x, y) = \{m_1, m_2, \cdots, m_t\}\}$ is an f-*monochromatic* rectangle. Hence, to minimize the number of messages, one needs to minimize the number of f-monochromatic rectangles which has led to research in this direction. Upper bounds are typically obtained by explicit constructions. For a comprehensive review, see Kushilevitz and Nisan (2006).

For control systems, the discussion takes further aspects into account including a design objective, system dynamics, and the uncertainty in the system variables.

Communication Complexity in Reach Control

Wong (2009) defines the communication complexity in networked control as follows: Consider a design specification where two DMs wish to steer the state of a dynamical system in finite time. This can be viewed as a setting in (1)–(2) with 4 DMs, where there is iterative communication between a sensor and a DM, and there is no stochastic noise in the system. Given a set of initial states $x_0 \in \mathcal{X}_0$, and finite sets of objective choices for each DM ($\mathcal{A}$ for DM 1, $\mathcal{B}$ for DM 2), the goal is to ensure that (i) there exists a finite time where both DMs know the final state of the system, (ii) the final state satisfies the choices of the DMs, and (iii) the finite time (when the objective is satisfied) is known by the DMs.

The communication complexity for such a system is defined as the infimum over all protocols of the supremum over the triple of initial states, and choices of the DMs, such that the above is satisfied. That is,

$$R(\mathcal{X}_0, \mathcal{A}, \mathcal{B}) = \inf_{\gamma, \underline{\mathcal{E}}} \sup_{\alpha, \beta, x_0} R(\gamma, \underline{\mathcal{E}}, \alpha, \beta, x_0),$$

where $R(\gamma, \underline{\mathcal{E}}, \alpha, \beta, x_0)$ denotes the communication rate under the control and coding functions $\gamma, \underline{\mathcal{E}}$, which satisfies the objectives given by the choices α, β and initial condition x_0.

Wong obtains a cut-set type lower bound: Given a fixed initial state, a lower bound is given by $2D(f)$, where f is a function of the objective choices and $D(f)$ is a variation of $R(f)$ introduced in (6) with the additional property that both DMs know f at the end of the protocol. An upper bound is established by the exchange of the initial states and objective functions also taking into account signaling, that is, the communication through control actions, which is discussed further below in the context of stabilization. Wong and Baillieul (2012) consider a detailed analysis for a real-valued bilinear controlled decentralized system.

Connections with Information Theory

Information theory literature has made significant contributions to such problems. An information theoretic setup typically entails settings where an unboundedly large sequence of messages are encoded and functions of which are to be computed. Such a setting is not applicable in a real-time setting but is very useful for obtaining performance bounds (i.e., good lower bounds on complexity) which can at certain instances be achievable even in a real-time setting. That is, instead of a single realization of random variables in the setup of (1)–(2), the average performance for a large number of independent realizations/copies for such problems is typically considered.

In such a context, Definitions 1 and 2 can be adjusted so that the communication complexity is computed by mutual information (Cover and Thomas 1991). Replacing the fixed-rate or variable-rate (entropy) constraint in Definition 1 with a mutual information constraint leads to convexity properties for $C(R)$ and $R(C)$. Such an information theoretic formulation can provide useful lower bounds and desirable analytical properties.

We note here the interesting discussion between decentralized computation and

communication provided by Orlitsky and Roche (2001) as well as by Witsenhausen (1976) where a *probability-free* construction is considered and a *zero-error (non-asymptotic and error-free)* computation is considered in the same spirit as in Yao (1979).

Such decentralized computation problems can be viewed as multiterminal source coding problems with a cost function aligned with the computation objective. Ma and Ishwar (2011) and Gamal and Kim (2012) provide a comprehensive treatment and review of information exchange requirements for computing. Essential in such constructions is the method of *binning*, which is a key tool in distributed source coding problems. Binning efficiently designates the enumeration of symbols (which can be confused in the absence of coding) given the relevant information at a receiver DM.

Such problems involve interactive communications as well as multiterminal coding problems. As mentioned earlier, it is also important to point out that multi-round protocols typically reduce the average rate requirements.

Communication Complexity in Decentralized Stabilization

An important relevant setting of reach control is where the target final state is the zero vector: The system is to be stabilized. Consider the following special case of (1)–(2) for an LTI system:

$$x_{t+1} = Ax_t + \sum_{j=1}^{L} B^j u_t^j,$$

$$y_t^i = C^i x_t \ t = 0, 1, \ldots \qquad (7)$$

where $i \in \mathcal{L}$, and it is assumed that the joint system is stabilizable and detectable, but the individual pairs (A, B^i) may not be stabilizable or (A, C^i) may not be detectable. Here, $x_t \in \mathbb{R}^n$ is the state, $u_t^i \in \mathbb{R}^{m_i}$ is the control applied by station i, and $y_t^i \in \mathbb{R}^{p_i}$ is the observation available at station i, all at time t. The initial state x_0 is generated according to a probability
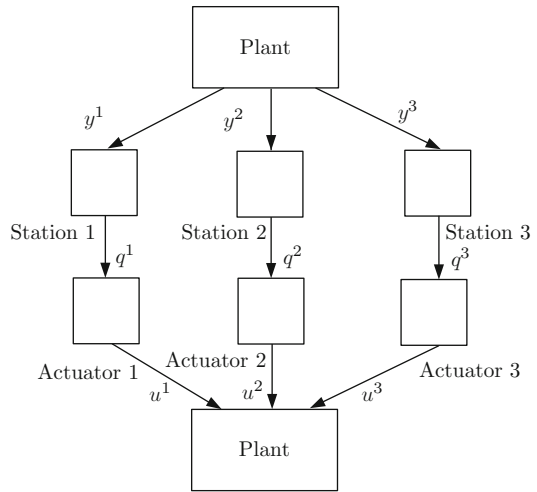
Information and Communication Complexity of Networked Control Systems, Fig. 2 Decentralized stabilization with multiple controllers

distribution supported on a compact set $\mathcal{X}_0 \subset \mathbb{R}^n$. We denote controllable and unobservable subspaces at station i by K^i and N^i and refer to the subspace orthogonal to N^i as the observable subspace at the ith station, denoted by O^i. The information available to station i at time t is $I_t^i = \{y_{[0,t]}^i, u_{[0,t-1]}^i\}$. For such a system (see Fig. 2), it is possible for the controllers to communicate through the plant with the process known as *signaling* which can be used for communication of mode information among the decision makers. Denote by $i \rightarrow j$ the property that DM i can signal to DM j. This holds if and only if $C^j (A)^l B^i \neq 0$, for at least one l, $1 \leq l \leq n$. A directed graph $\mathcal{G}$ among the L stations can be constructed through such a communication relationship.

Suppose that A is such that in its Jordan form, where each Jordan block admits distinct real eigenvalues. Then, a lower bound on the communication complexity (per time stage) for stabilizability is given by $\sum_{|\lambda_i|>1} \eta_{M_i} \log_2(|\lambda_i|)$, where

$$\eta_{M_i} = \min_{l,m \in \{1,2,\ldots,L\}} \{d(l,m) + 1 : l \rightarrow m,$$

$$[x^i] \subset O^i \cup O^m, \quad [x^i] \subset K^m\},$$

with $d(l, m)$ denoting the graph distance (number of edges in a shortest path) between DM l and DM m in $\mathcal{G}$ and $[x_i]$ denoting the subspace spanned by x_i. Furthermore, there exist stabilizing coding and control policies whose sum rate is arbitrarily close to this bound. When different Jordan blocks may admit repeated and possibly complex eigenvalues, variations of the result above are applicable. In the special case where there is a centralized controller which receives information from multiple sensors (under stabilizability and joint detectability), even in the presence of noise, to achieve asymptotic stability, it suffices to have the average total rate be greater than $\sum_{|\lambda_i| > 1} \log_2(|\lambda_i|)$. The results above follow from Matveev and Savkin (2008) and Yüksel and Başar (2013). For the case with a single sensor, this result has been studied extensively in networked control (see the chapter on ▶ Quantized Control and Data Rate Constraints in the Encyclopedia).

Summary and Future Directions

In this text, we discussed the problem of communication complexity in networked control systems. Our analysis considered both cost minimization and controllability/reachability problems subject to information constraints. We also discussed the communication complexity in distributed computing as has been studied in the computer science community and provided a brief discussion on the information theoretic approaches for such problems together with structural results. There are many relevant open problems on structural results for optimal policies, explicit solutions, as well as nontrivial upper and lower bounds on the optimal performance.

Cross-References

▶ Data Rate of Nonlinear Control Systems and Feedback Entropy
▶ Flocking in Networked Systems
▶ Information-Based Multi-Agent Systems

▶ Networked Control Systems: Estimation and Control over Lossy Networks
▶ Quantized Control and Data Rate Constraints

Recommended Reading

The information exchange requirements for decentralized optimization depend also on the structural properties of the cost functional to be minimized. For a class of team problems, one might simply need to exchange a sufficient statistic needed for optimal solutions. For some problems, there may be no need for an exchange at all, if the sufficient statistics are already available, as in the case of *mean field equilibrium* problems when the number of decision makers is unbounded or very large for almost optimal solutions; see Huang et al. (2006) and Lasry and Lions (2007). In case there is no common probabilistic information, the problem considered becomes further involved. The consensus literature, under both Bayesian and non-Bayesian contexts, aims to achieve agreement on a class of system variables under information constraints: see, e.g., Tsitsiklis et al. (1986). Optimization under local interaction and sparsity constraints and various criteria have been investigated in a number of publications including Rotkowitz and Lall (2006). A review for the literature on norm-optimal control as well as optimal stochastic dynamic teams is provided in Mahajan et al. (2012). Tsitsiklis and Athans (1985) have observed that from a computational complexity viewpoint, obtaining optimal solutions for a class of such communication protocol design problems is non-tractable (NP-hard).

Even though obtaining explicit solutions for optimal coding and control results may be difficult, it is useful to obtain structural results on optimal coding and control policies since one can reduce the search space to a smaller class of functions. For dynamic team problems, these typically follow from the construction of a controlled Markov chain (see Walrand and Varaiya 1983) and applying tools from stochastic control theory which obtain structural results on optimal coding and control policies (see Nayyar et al. 2013).

Along these lines, for system (1)–(2), if the DMs can agree on a joint belief $P(x_t \in \cdot | I_t^i, i \in \mathcal{L})$ at every time stage, then the optimal cost that would be achieved under a centralized system could be attained (see Yüksel and Başar 2013). As a further important illustrative case, if the problem described in Definition 1 is for a real-time estimation problem for a Markov source, then the optimal causal fixed-rate coder minimizing any cost function uses only the last source symbol and the information at the controller's memory: see Witsenhausen (1979). We also note that the optimal design of information channels for optimization under information constraints is a non-convex problem; see Yüksel and Linder (2012) and Yüksel and Başar (2013) for a review of the literature and certain topological properties of the problem. We refer the reader to Nemirovsky and Yudin (1983) for a comprehensive resource on information complexity for optimization problems. A sequential setting with an information theoretic approach to the formulation of communication complexity has been considered in Raginsky and Rakhlin (2011). A formulation relevant to the one in Definition 1 has been considered in Teneketzis (1979) with mutual information constraints. Giridhar and Kumar (2006) discuss distributed computation for a class of symmetric functions under information constraints and present a comprehensive review.

Bibliography

Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York

Gamal AE, Kim YH (2012) Network information theory. Cambridge University Press, UK

Giridhar A, Kumar P (2006) Toward a theory of in-network computation in wireless sensor networks. IEEE Commun Mag 44:98–107

Huang M, Caines PE, Malhamé RP (2006) Large population stochastic dynamic games: closed-loop McKean-vlasov systems and the nash certainty equivalence principle. Commun Inf Syst 6: 221–251

Kushilevitz E, Nisan N (2006) Communication complexity, 2nd edn. Cambridge University Press, New York

Lasry JM, Lions PL (2007) Mean field games. Jpn J Math 2:229–260

Ma N, Ishwar P (2011) Some results on distributed source coding for interactive function computation. IEEE Trans Inf Theory 57:6180–6195

Mahajan A, Martins N, Rotkowitz M, Yüksel S (2012) Information structures in optimal decentralized control. In: IEEE conference on decision and control, Hawaii

Matveev AS, Savkin AV (2008) Estimation and control over communication networks. Birkhäuser, Boston

Nayyar A, Mahajan A, Teneketzis D (2013) The common-information approach to decentralized stochastic control. In: Como G, Bernhardsson B, Rantzer A (eds) Information and control in networks. Springer International Publishing, Switzerland

Nemirovsky A, Yudin D (1983) Problem complexity and method efficiency in optimization. Wiley-Interscience, New York

Orlitsky A, Roche JR (2001) Coding for computing. IEEE Trans Inf Theory 47:903–917

Raginsky M, Rakhlin A (2011) Information-based complexity, feedback and dynamics in convex programming. IEEE Trans Inf Theory 57: 7036–7056

Rotkowitz M, Lall S (2006) A characterization of convex problems in decentralized control. IEEE Trans Autom Control 51:274–286

Teneketzis D (1979) Communication in decentralized control. PhD dissertation, MIT

Tsitsiklis J, Athans M (1985) On the complexity of decentralized decision making and detection problems. IEEE Trans Autom Control 30:440–446

Tsitsiklis J, Bertsekas D, Athans M (1986) Distributed asynchronous deterministic and stochastic gradient optimization algorithms. IEEE Trans Autom Control 31:803–812

Walrand JC, Varaiya P (1983) Optimal causal coding-decoding problems. IEEE Trans Inf Theory 19:814–820

Witsenhausen HS (1976) The zero-error side information problem and chromatic numbers. IEEE Trans Inf Theory 22:592–593

Witsenhausen HS (1979) On the structure of real-time source coders. Bell Syst Tech J 58:1437–1451

Wong WS (2009) Control communication complexity of distributed control systems. SIAM J Control Optim 48:1722–1742

Wong WS, Baillieul J (2012) Control communication complexity of distributed actions. IEEE Trans Autom Control 57:2731–2345

Yao ACC (1979) Some complexity questions related to distributive computing. In: Proceedings of the 11th annual ACM symposium on theory of computing, Atlanta

Yüksel S, Başar T (2013) Stochastic networked control systems: stabilization and optimization under information constraints. Birkhäuser, Boston

Yüksel S, Linder T (2012) Optimization and convergence of observation channels in stochastic control. SIAM J Control Optim 50:864–887

Information Structures, the Witsenhausen Counterexample, and Communicating Using Actions

Pulkit Grover
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

The concept of "information structures" in decentralized control is a formalization of the notion of "who knows what and when do they know it." Even seemingly simple problems with simply stated information structures can be extremely hard to solve. Perhaps the simplest of such unsolved problem is the celebrated Witsenhausen counterexample, formulated by Hans Witsenhausen in 1968. This entry discusses how the information structure of the Witsenhausen counterexample makes it hard and how an information-theoretic approach, which explores the knowledge gradient due to a nonclassical information pattern, has helped obtain insights into the problem.

Keywords

Decentralized control; Information theory; Implicit communication; Team decision theory

Introduction

Modern control systems often comprise of multiple decentralized control agents that interact over communication channels (Fig. 1). What characteristic distinguishes a centralized control problem from a decentralized one? One fundamental difference is a "knowledge gradient": agents in a decentralized team often observe, and hence know, different things. This observation leads to the idea of *information patterns* (Witsenhausen 1971), a formalization of the notion of "who

knows what and when do they know it" (Ho et al. 1978; Mitter and Sahai 1999).

The information pattern is said to be *classical* if all agents in the team receive the same information and have perfect recall (so they do not forget it). What is so special about classical information patterns? For these patterns, the presence of external communication links has no effect on the optimal costs! After all, what could the agents use the communication links for, when there is no knowledge gradient? More interesting, therefore, are the problems for which the information pattern is *non*classical. These problems sit at the intersection of communication and control: *communication between agents* can help reduce the knowledge differential that exists between them, helping them perform the *control* task. Intellectually and practically, the concept of nonclassical information patterns motivates a lot of formulations at the control-communication intersection. Many of these formulations – including some discussed in this Encyclopedia (e.g., ▸ Data Rate of Nonlinear Control Systems and Feedback Entropy; ▸ Information and Communication Complexity of Networked Control Systems ▸ Quantized Control and Data Rate Constraints; ▸ Networked Control Systems: Architecture and Stability Issues; and ▸ Networked Control Systems: Estimation and Control Over Lossy Networks) – intellectually ask the question: for a realistic channel that is constrained by noise, bandwidth, and speed, what is the optimal communication *and* control strategy?

One could ask the question of optimal control strategy even for decentralized control problems where no external channel is available to bridge this knowledge gradient. Why could these problems be of interest? First, these problems are limiting cases of control with communication constraints. Second, and perhaps more importantly, they bring out an interesting possibility that can allow the agents to "communicate," i.e., exchange information, *even when the external channel is absent*. It is possible to use control actions to *communicate* through changing the system state! We now introduce this form of communication using a simple toy example.
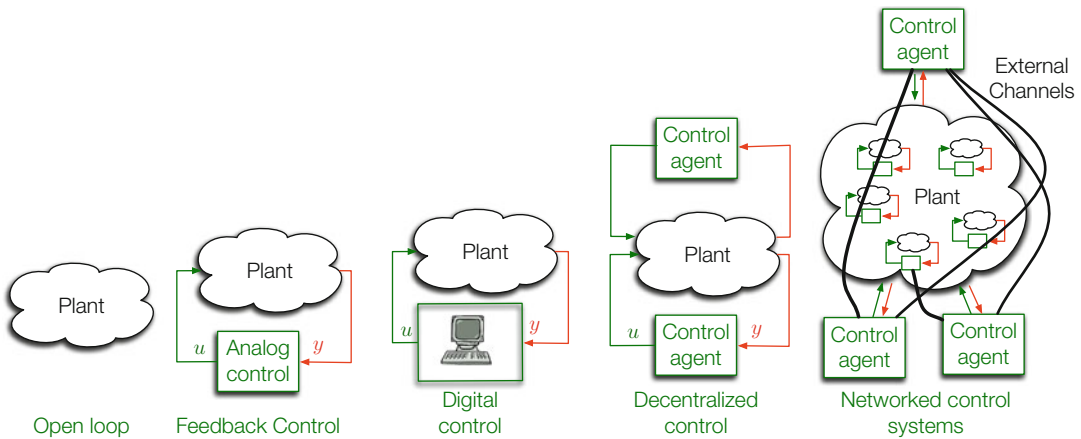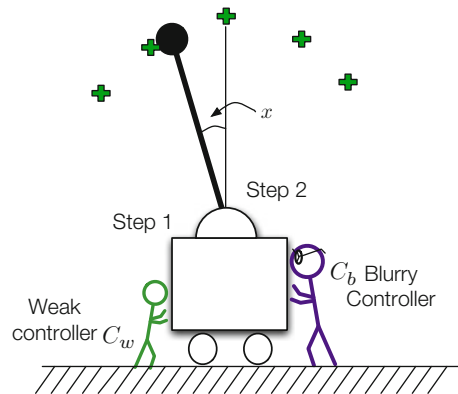
Information Structures, the Witsenhausen Counterexample, and Communicating Using Actions, Fig. 1 The evolution of control systems. Modern "net-worked control systems" (also called "cyber-physical systems") are decentralized and networked using communication channels

Communicating Using Actions: An Example

To gain intuition into when communication using actions could be useful, consider the inverted pendulum example shown in Fig. 2. The goal of the two agents is to bring the pendulum as close to the origin as possible. Both controllers have their strengths and weaknesses. The "weak" controller C_w has little energy, but has perfect state observations. On the other hand, the "blurry" controller C_b has infinite energy, but noisy observations. They act one after the other, and their goal is to move the pendulum close to the center from any initial state. The information structure of the problem is nonclassical: the C_w, but not C_b, knows the initial state of the pendulum, and C_w does not know the precise (noisy) observation of C_b using which C_b takes actions.

A possible strategy: A little thought reveals an interesting strategy – the weak controller, having perfect observations, can move the state to the closest of some predecided points in space, effectively *quantizing* the state. If these quantization points are sufficiently far from each other, they can be estimated accurately (through the noise) by the blurry controller, which can then use its energy to push the pendulum all the way to zero. In this way, the weak controller expends little energy, but is able to "communicate" the state through the noise to the blurry controller, by
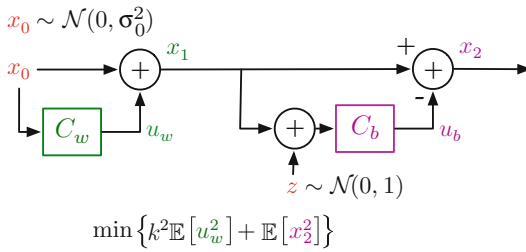

Information Structures, the Witsenhausen Counterexample, and Communicating Using Actions, Fig. 2 Two controllers, with their respective strengths and weaknesses, attempting to bring an inverted pendulum close to the center. Also shown (using green "+" signs) are possible quantization points chosen by the controllers for a quantization-based control strategy

making it take values on a finite set. Once the blurry controller has received the state through the noise, it can use its infinite energy to push the state to zero.

The Witsenhausen Counterexample

The above two-controller inverted-pendulum example is, in fact, motivated by what is now known as "the Witsenhausen counterexample," formulated by Witsenhausen in 1968 (see Fig. 3).

$$x_0 \sim \mathcal{N}(0, \sigma_0^2)$$

$$\min \left\{ k^2 \mathbb{E}\left[u_w^2 \right] + \mathbb{E}\left[x_2^2 \right] \right\}$$
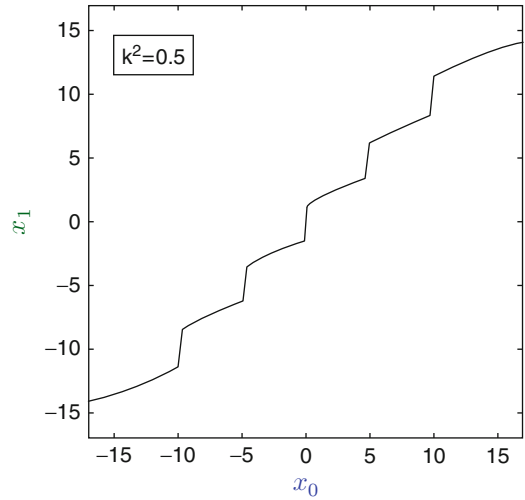
Information Structures, the Witsenhausen Counterexample, and Communicating Using Actions, Fig. 3 The Witsenhausen counterexample is a deceptively simple two-time-step two-controller decentralized control problem. The weak and the blurry controllers, C_w and C_b act in a sequential manner

Information Structures, the Witsenhausen Counterexample, and Communicating Using Actions, Fig. 4 The optimization solution of Baglietto et al. (1997) for $k^2 = 0.5$, $\sigma_0^2 = 5$. The information-theoretic strategy of "dirty-paper coding" Costa (1983) also yields the same curve (Grover and Sahai 2010)

In the counterexample, two controllers (denoted here by C_w for "weak" and C_b for "blurry") act one after the other in two time-steps to minimize a quadratic cost function. The system state is denoted by x_t, where t is the time index. u_w and u_b denote the inputs generated by the two controllers. The cost function is $k^2 \mathbb{E}\left[u_w^2 \right] + \mathbb{E}\left[x_2^2 \right]$ for some constant k. The initial state x_0 and the noise z at the input of the blurry controller are assumed to be Gaussian distributed and independent, with variances σ_0^2 and 1 respectively. The problem is a "linear-quadratic-Gaussian" (LQG) problem, i.e., the state evolution is linear, the costs are quadratic, and the primitive random variables are Gaussian.

Why is the problem called a "counterexample"? The traditional "certainty-equivalence" principle (Bertsekas 1995) shows that for all centralized LQG problems, linear control laws are optimal. Witsenhausen (1968) provided a nonlinear strategy for the Witsenhausen problem which outperforms all linear strategies. Thus, the counterexample showed that the certainty-equivalence doctrine does not extend to decentralized control.

What is this strategy of Witsenhausen that outperforms all linear strategies? It is, in fact, a quantization-based strategy, as suggested in our inverted-pendulum story above. Further, it was shown by Mitter and Sahai (1999) that multipoint quantization strategies can outperform linear strategies by an arbitrarily large factor! This observation, combined with the simplicity of the counterexample, makes the problem very

important in decentralized control. This simple two-time-step two-controller LQG problem needs to be understood to have any hope of understanding larger and more complex problems.

While the optimal costs for the problem are still unknown (even though it is known that an optimal strategy exists (Witsenhausen 1968; Wu and Verdú 2011)), there exists a wealth of understanding of the counterexample that has helped address more complicated problems. A body of work, starting with that of Baglietto et al. (1997), numerically obtained solutions that could be close to optimal (although there is no mathematical proof thereof). All these solutions have a consistent form (illustrated in Fig. 4), with slight improvements in the optimal cost. Because the discrete version of the problem, appropriately relaxed, is known to be NP-complete (Papadimitriou and Tsitsiklis 1986), this approach cannot be used to understand the entire parameter space and hence has focused on one point: $k^2 = 0.5, \sigma_0^2 = 5$. Nevertheless, the approach has proven to be insightful: a recent information-theoretic body of work shows that the strategies of Fig. 4 can be thought of as information-theoretic strategies of "dirty-paper coding" Costa (1983) that is related to the idea of

embedding information in the state. The question here is: how do we embed the information about the state *in the state itself*?

An information-theoretic view of the counterexample: This information-theoretic approach that culminated in Grover et al. (2013) also obtained the first approximately optimal solutions to the Witsenhausen counterexample as well as its vector extensions. The result is established by analyzing information flows in the counterexample that work toward minimizing the knowledge gradient, effectively an information pattern in which C_w can predict the observation of C_b more precisely. The analysis provides *an information-theoretic lower bound on cost that holds irrespective of what strategy is used*. For the original problem, this characterizes the optimal costs (with associated strategies) within a factor of 8 for all problem parameters (i.e., k and σ_0^2). For any finite-length extension, uniform finite-ratio approximations also exist (Grover et al. 2013). The asymptotically infinite-length extension has been solved *exactly* (Choudhuri and Mitra 2012).

The problem has also driven delineation of decentralized LQG control problems with optimal linear solutions and those with nonlinear optimal solutions. This led to the development and understanding of many variations of the counterexample (Bansal and Başar 1987; Başar 2008; Ho et al. 1978; Rotkowitz 2006) and understanding that can extend to larger decentralized control problems. More recent work shows that the promise of the Witsenhausen counterexample was not a misplaced one: the information-theoretic approach that provides approximately optimal solutions to the counterexample (Grover et al. 2013) yields solutions to other more complex (e.g., multi-controller, more time-steps) problems as well (Grover 2010; Park and Sahai 2012).

by the Witsenhausen counterexample. However, nonclassical information pattern for some simple problems – starting with the counterexample – has recently been explored via an information-theoretic lens, yielding the first optimal or approximately optimal solutions to these problems. This approach is promising for larger decentralized control problems as well. It is now important to explore what is the simplest decentralized control problem that cannot be solved (exactly or approximately) using ideas developed for the counterexample. In this manner, the Witsenhausen counterexample can provide a platform to unify the more modern (i.e., external-channel centric approaches, see ▶ Quantized Control and Data Rate Constraints; ▶ Data Rate of Nonlinear Control Systems and Feedback Entropy; ▶ Networked Control Systems: Architecture and Stability Issues; ▶ Networked Control Systems: Estimation and Control Over Lossy Networks; ▶ Information and Communication Complexity of Networked Control Systems; in the encyclopedia) with the more classical decentralized LQG problems, leading to enriching and useful formulations.

Cross-References

- ▶ Data Rate of Nonlinear Control Systems and Feedback Entropy
- ▶ Information and Communication Complexity of Networked Control Systems
- ▶ Networked Control Systems: Architecture and Stability Issues
- ▶ Networked Control Systems: Estimation and Control over Lossy Networks
- ▶ Quantized Control and Data Rate Constraints

Summary and Future Directions

Even simple problems with nonclassical information structures can be hard to solve using classical techniques, as is demonstrated

Bibliography

Baglietto M, Parisini T, Zoppoli R (1997) Nonlinear approximations for the solution of team optimal control problems. In: Proceedings of the IEEE

conference on decision and control (CDC), San Diego, pp 4592–4594

Bansal R, Başar T (1987) Stochastic teams with non-classical information revisited: when is an affine control optimal? IEEE Trans Autom Control 32:554–559

Başar T (2008) Variations on the theme of the Witsenhausen counterexample. In: Proceedings of the 47th IEEE conference on decision and control (CDC), Cancun, pp 1614–1619

Bertsekas D (1995) Dynamic programming. Athena Scientific, Belmont

Costa M (1983) Writing on dirty paper. IEEE Trans Inf Theory 29(3):439–441

Choudhuri C, Mitra U (2012) On Witsenhausen's counterexample: the asymptotic vector case. In: IEEE information theory workshop (ITW), Lausanne, pp 162–166

Grover P (2010) Actions can speak more clearly than words. Ph.D. dissertation, UC Berkeley, Berkeley

Grover P, Sahai A (2010) Vector Witsenhausen counterexample as assisted interference suppression. Int J Syst Control Commun 2:197–237. Special issue on Information Processing and Decision Making in Distributed Control Systems

Grover P, Park S-Y, Sahai A (2013) Approximately-optimal solutions to the finite-dimensional Witsenhausen counterexample. IEEE Trans Autom Control 58(9):2189

Ho YC, Kastner MP, Wong E (1978) Teams, signaling, and information theory. IEEE Trans Autom Control 23(2):305–312

Mitter SK, Sahai A (1999) Information and control: Witsenhausen revisited. In: Yamamoto Y, Hara S (eds) Learning, control and hybrid systems. Lecture notes in control and information sciences, vol 241. New York, NY: Springer, pp 281–293

Papadimitriou CH, Tsitsiklis JN (1986) Intractable problems in control theory. SIAM J Control Optim 24(4):639–654

Park SY, Sahai A (2012) It may be easier to approximate decentralized infinite-horizon LQG problems. In: 51st IEEE Conference on Decision and Control (CDC), Maui, pp 2250–2255

Rotkowitz M (2006) Linear controllers are uniformly optimal for the Witsenhausen counterexample. In: Proceedings of the 45th IEEE conference on decision and control (CDC), San Diego, pp 553–558

Witsenhausen HS (1968) A counterexample in stochastic optimum control. SIAM J Control 6(1):131–147

Witsenhausen HS (1971) Separation of estimation and control for discrete time systems. Proc IEEE 59(11):1557–1566

Wu Y, Verdú S (2011) Witsenhausen's counterexample: a view from optimal transport theory. In: IEEE conference on decision and control and European control conference (CDC-ECC), Orlando, pp 5732–5737

Information-Based Multi-Agent Systems

Wing-Shing Wong[1] and John Baillieul[2]
[1]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China
[2]Mechanical Engineering, Electrical and Computer Engineering, Boston University, Boston, MA, USA

Abstract

Multi-agent systems are encountered in nature (animal groups), in various domains of technology (multi-robot networks, mixed robot-human teams) and in various human activities (such as dance and team athletics). Information exchange among agents ranges from being incidentally important to crucial in such systems. Several systems in which information exchange among the agents is either a primary goal or a primary enabler are discussed briefly. Specific topics include power management in wireless communication networks, data-rate constraints, the complexity of distributed control, robotics networks and formation control, action-mediated communication, and multi-objective distributed systems.

Keywords

Distributed control; Information constraints; Multi-agent systems

Introduction

The role of information patterns in the decentralized control of multi-agent systems has been studied in different theoretical contexts for more than five decades. The paper Ho (1972) provides references to early work in this area. While research on distributed decision making has continued, a large body of recent research on robotic networks has brought new dimensions of geometric aspects of information patterns to the

forefront (Bullo et al. 2009). At the same time, machine intelligence, machine learning, machine autonomy, and theories of operation of mixed teams of humans and robots have considerably extended the intellectual frontiers of information-based multi-agent systems (Baillieul et al. 2012). A further important development has been the study of action-mediated communication and the recently articulated theory of *control communication complexity* (Wong and Baillieul 2012). These developments may shed light on nonverbal forms of communication among biological organisms (including humans) and on the intrinsic energy requirements of information processing.

In conventional decentralized control, the control objective is usually well-defined and known to all agents. Multi-agent information-based control encompasses a broader scenario, where the objective can be agent dependent and is not necessarily explicitly announced to all. For illustration, consider the power control problem in wireless communication – one of the earliest engineering systems that can be regarded as multi-agent information based. It is common that multiple transmitter-receiver communication pairs share the same radio frequency band and the transmission signals interfere with each other. The power control problem searches for feedback control for each transmitter to set its power level. The goal is for each transmitter to achieve targeted signal-to-interference ratio (SIR) level by using information of the observed levels at the intended receiver only.

A popular version of the power control problem (Foschini and Miljanic 1993) defines each individual objective target level by means of a requirement threshold, known only to the intended transmitter. As SIR measurements naturally reside on a receiver, the observed SIR needs to be communicated back to the transmitter. For obvious reasons, the bandwidth for such communication is limited. The resulting model fits the bill of multi-agent information-based control. In Sung and Wong (1999), a tristate power control strategy is proposed so that the power control outputs are either increased or decreased by a fixed dB or no change at all. Convergence of the feedback

algorithm was shown using a Lyapunov-like function.

This entry surveys key topics related to multi-agent information-based control systems, including control complexity, control with data-rate constraints, robotic networks and formation control, action-mediated communication, and multi-objective distributed systems.

Control Complexity

In information-based distributed control systems, how to efficiently share computational and communication resources is a fundamental issue. One of the earliest investigations on how to schedule communication resources to support a network of sensors and actuators is discussed in Brockett (1995). The concept of *communication sequencing* was introduced to describe how the communication channel is utilized to convey feedback control information in a network consisting of interacting subsystems. In Brockett (1997), the concept of *control attention* was introduced to provide a measure of the complexity of a control law against its performance. As attention is a shared, limited resource, the goal is to find minimum attention control. Another approach to gauge control complexity in a distributed system is by means of the minimum amount of communicated data required to accomplish a given control task.

Control with Data-Rate Constraints

A fundamental challenge in any control implementation in which system components communicate with each other over communication links is ensuring that the channel capacity is large enough to deal with the fastest time constants among the system components. In a single agent system, the so-called Data-Rate Theorem has been formulated in various ways to understand the constraints imposed between the sensor and the controller and between the controller and the actuator. Extensions to this fundamental result have been focused on addressing similar

problems in the network control system context. Information on such extensions in the distributed control setting can be found in Nair and Evans (2004) and Yüksel and Basar (2007).

Robotic Networks and Formation Control

The defining characteristic of robotic networks within the larger class of multi-agent systems is the centrality of spatial relationships among network nodes. Graph theory has been shown to provide a generally convenient mathematical language in which to describe spatial concepts and it is the key to understanding spatial rigidity related to the control of formations of autonomous vehicles (Anderson et al. 2008), or in flocking systems (Leonard et al. 2012), or in consensus problems (Su and Huang 2012), or in rendezvous problems (Cortés et al. 2006). For these distributed control research topics, readers can consult other sections in this Encyclopedia for a comprehensive reference list.

Much of the recent work on formation control has included information limitation considerations. For consensus problems, for example, Olfati-Saber and Murray (2004) introduced a sensing cost constraint, and in Ren and Beard (2005) information exchange constraints are considered, and in Yu and Wang (2010) communication delays are explicitly modeled.

Action-Mediated Communication

Biological organisms communicate through motion. Examples of this include prides of lions or packs of wolves whose pursuit of prey is a cooperative effort and competitive team athletics in the case of humans. Recent research has been aimed at developing a theoretical foundation of action-mediated communication. Communication protocols for motion-based signaling between mobile robots have been developed (Raghunathan and Baillieul 2009) and preliminary steps towards a theory of artistic expression through controlled

movements in dance have been reported in Baillieul and Özcimder (2012). Motion-based communication of this type involves specially tailored motion description languages in which sequences of motion primitives are assembled with the objective of conveying artistic intent, while minimizing the use of limited energy resources in carrying out the movement. These motion primitives constitute the *alphabet* that enables communication, and physical constraints on the motions define the grammatical rules that govern the ways in which motion sequences may be constructed.

Research on action-mediated communication helps illustrate the close connection between control and information theory. Further discussion of the deep connection between the two can be found, for example, in Park and Sahai (2011), which argues for the equivalence between the stabilization of a distributed linear system and the capacity characterization in linear network coding.

Multi-objective Distributive Systems

In a multi-agent system, agents may aim to carry out individual objectives. These objectives can either be cooperatively aligned (such as in a cooperative control setting) or may contend antagonistically (such as in a zero-sum game setting). In either case, a common assumption is that the objective functions are a priori known to all agents. However, in many practical applications, agents do not know the objectives of other agents, at least not precisely. For example, in the power control problem alluded to earlier, the signal-to-interference requirement of a user may be unknown to other users. Yet this does not prevent the possibility of deriving convergence algorithms to allow the joint goals to be achieved.

The issue of unknown objectives in a multi-agent system is formally analyzed in Wong (2009) via the introduction of choice-based actions. In an open access network, objectives of an individual agent may be known only partially, via the form of a random distribution in some cases. In order to achieve a joint control objective in general, some communication via the system

is required if there is no side communications channel. A basic issue is how to measure the minimum amount of information exchange that is required to perform a specific control task. Motivated by the idea of communication complexity in computer science, the idea of control communication complexity was introduced in Wong (2009), which can provide such a measure. In Wong and Baillieul (2009), the idea was extended to a rich class of nonlinear systems that arise as models of physical processes ranging from rigid body mechancs to quantum spin systems.

In some special cases, control objectives can be achieved without any communication among the agents. For systems with bilinear input–output mapping, including the Brockett Integrator, it is possible to derive conditions that guarantee this property (Wong and Baillieul 2012). Moreover, for quadratic type of control cost, it is possible to compute the optimal control cost. Similar results can be extended to linear systems as discussed in Liu et al. (2013). This circle of ideas is connected to the so-called *standard parts* problem as investigated in Baillieul and Wong (2009). Another connection is to *correlated equilibrium* problems that have been recently studied by game theorists Shoham and Leyton-Brown (2009).

Cross-References

▶ Motion Description Languages and Symbolic Control
▶ Multi-vehicle Routing
▶ Networked Systems

Bibliography

Altman E, Avrachenkov K, Menache I, Miller G, Prabhu BJ, Shwartz A (2009) Dynamic discrete power control in cellular networks. IEEE Trans Autom Control 54(10):2328–2340

Anderson B, Yu C, Fidan B, Hendrickx J (2008) Rigid graph control architectures for autonomous formations. IEEE Control Syst Mag 28(6):48–63. doi:10.1109/MCS.2008.929280

Baillieul J, Özcimder K (2012) The control theory of motion-based communication: problems in teaching robots to dance. In: Proceedings of the American control conference (ACC), Montreal, 27–29 June 2012, pp 4319–4326

Baillieul J, Wong WS (2009) The standard parts problem and the complexity of control communication. In: Proceedings of the 48th IEEE conference on decision and control, held jointly with the 28th Chinese control conference, Shanghai, 16–18 Dec 2009, pp 2723–2728

Baillieul J, Leonard NE, Morgansen KA (2012) Interaction dynamics: the interface of humans and smart machines. Proc IEEE 100(3):776–801. doi:10.1109/JPROC.s011.2180055

Brockett RW (1995) Stabilization of motor networks. In: Proceedings of the 34th IEEE conference on decision and control, New Orleans, 13–15 Dec 1995, pp 1484–1488

Brockett RW (1997) Minimum attention control. In: Proceedings of the 36th IEEE conference on decision and control, San Diego, 10–12 Dec 1997, pp 2628–2632

Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks: a mathematical approach to motion coordination algorithms. Princeton University Press, Princeton

Cortés J, Martínez S, Bullo F (2006) Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Trans Autom Control 51(8):1289–1298

Foschini GJ, Miljanic Z (1993) A simple distributed autonomous power control algorithm and its convergence. IEEE Trans Veh Technol 42(4):641–646

Ho YC (1972) Team decision theory and information structures in optimal control problems-part I. IEEE Trans Autom Control 17(1):15–22

Leonard NE, Young G, Hochgraf K, Swain D, Chen W, Marshall S (2012) In the dance studio: analysis of human flocking. In: Proceedings of the 2012 American control conference, Montreal, 27–29 June 2012, pp 4333–4338

Liu ZC, Wong WS, Guo G (2013) Cooperative control of linear systems with choice actions. In: Proceedings of the American control conference, Washington, DC, 17–19 June 2013, pp 5374–5379

Nair GN, Evans RJ (2004) Stabilisability of stochastic linear systems with finite feedback data rates. SIAM J Control Optim 43(2):413–436

Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans Autom Control 49(9):1520–1533

Park SY, Sahai A (2011) Network coding meets decentralized control: capacity-stabilizability equivalence. In: Proceedings of the 50th IEEE conference on decision and control and European control conference, Orlando, 12–15 Dec 2011, pp 4817–4822

Raghunathan D, Baillieul J (2009) Motion based communication channels between mobile robots-A novel paradigm for low bandwidth information exchange. In: Proceedings of the IEEE/RSJ international conference

on intelligent robots and systems, 11–15 Oct 2009,
St. Louis, pp 702–708

Ren W, Beard RW (2005) Consensus seeking in multia-
gent systems under dynamically changing interaction
topologies. IEEE Trans Autom Control 50(5):655–661

Shoham Y, Leyton-Brown K (2009) Multiagent systems:
algorithmic, game-theoretic, and logical foundations.
Cambridge University Press, New York. ISBN:978-0-
521-89943-7

Su Y, Huang J (2012) Two consensus problems for
discrete-time multi-agent systems with switching net-
work topology. Automatica 48(9):1988–1997

Sung CW, Wong WS (1999) A distributed fixed-step
power control algorithm with quantization and ac-
tive link quality protection. IEEE Trans Veh Technol
48(2):553–562

Wong WS (2009) Control communication complexity of
distributed control systems. SIAM J Optim Control
48(3):1722–1742

Wong WS, Baillieul J (2009) Control communication
complexity of nonlinear systems. Commun Inf Syst
9(1):103–140

Wong WS, Baillieul J (2012) Control communication
complexity of distributed actions. IEEE
Trans Autom Control 57(11):2731–2745.
doi:10.1109/TAC.2012.2192357

Yu J, Wang L (2010) Group consensus in multi-agent
systems with switching topologies and communication
delays. Syst Control Lett 59(6):340–348

Yüksel S, Basar T (2007) Optimal signaling policies
for decentralized multicontroller stabilizability over
communication channels. IEEE Trans Autom Control
52(10):1969–1974

Input-to-State Stability

Eduardo D. Sontag
Rutgers University, New Brunswick, NJ, USA

Synonyms

ISS

Abstract

The notion of input to state stability (ISS)
qualitatively describes stability of the mapping
from initial states and inputs to internal states
(and more generally outputs). This entry focuses
on the definition of ISS and a discussion of
equivalent characterizations.

Keywords

Asymptotic stability; Dissipation; Lyapunov
functions

Introduction

We consider here systems with inputs in the usual
sense of control theory:

$$\dot{x}(t) = f(x(t), u(t))$$

(the arguments "t" are often omitted). There
are n state variables and m input channels.
States $x(t)$ take values in Euclidean space
$\mathbb{R}^n$, and the inputs (also called "controls"
or "disturbances" depending on the context)
are measurable in locally essentially bounded
maps $u(\cdot) : [0, \infty) \to \mathbb{R}^m$. The map $f :
\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be locally
Lipschitz with $f(0, 0) = 0$. The solution, defined
on some maximal interval $[0, t_{\max}(x^0, u))$, for
each initial state x^0 and input u, is denoted as
$x(t, x^0, u)$ and, in particular, for systems with
no inputs $\dot{x}(t) = f(x(t))$, just as $x(t, x^0)$. The
zero system associated to $\dot{x} = f(x, u)$ is by
definition the system with no inputs $\dot{x} = f(x, 0)$.
Euclidean norm is written as $|x|$. For a function
of time, typically an input or a state trajectory,
$\|u\|$, or $\|u\|_\infty$ for emphasis, is the (essential)
supremum or "sup" norm (possibly $+\infty$, if u is
not bounded). The norm of the restriction of a
signal to an interval I is denoted by $\|u_I\|_\infty$ (or
just $\|u_I\|$).

Input-to-State Stability

It is convenient to introduce "comparison func-
tions" to quantify stability. A *class $\mathcal{K}_\infty$ function*
is a function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ which is con-
tinuous, strictly increasing, and unbounded and
satisfies $\alpha(0) = 0$, and a *class $\mathcal{KL}$ function* is
a function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that
$\beta(\cdot, t) \in \mathcal{K}_\infty$ for each t and $\beta(r, t)$ decreases to
zero as $t \to \infty$, for each fixed r.

For a system with no inputs $\dot{x} = f(x)$, there is a well-known notion of global asymptotic stability (for short from now on, *GAS*, or "*0-GAS*" when referring to the zero system $\dot{x} = f(x, 0)$ associated to a given system with inputs $\dot{x} = f(x, u)$) due to Lyapunov and usually defined in "ϵ-δ" terms. It is an easy exercise to show that this standard definition is in fact equivalent to the following statement:

$$(\exists \beta \in \mathcal{KL}) |x(t, x^0)| \le \beta\left(|x^0|, t\right) \forall x^0, \ \forall t \ge 0.$$

The notion of input to state stability (ISS) was introduced in Sontag (1989), and it provides theoretical concepts used to describe stability features of a mapping $(u(\cdot), x(0)) \alpha x(\cdot)$ that sends initial states and input functions into states (or, more generally, outputs). Prominent among these features are that inputs that are bounded, "eventually small," "integrally small," or convergent should lead to outputs with the respective property. In addition, ISS and related notions quantify in what manner initial states affect transient behavior. The formal definition is as follows.

A system is said to be *input to state stable (ISS)* if there exist some $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that

$$|x(t)| \le \beta\left(|x^0|, t\right) + \gamma\left(\|u\|_\infty\right) \quad \text{(ISS)}$$

holds for all solutions (meaning that the estimate is valid for all inputs $u(\cdot)$, all initial conditions x^0, and all $t \ge 0$). Note that the supremum $\sup_{s \in [0,t]} \gamma(|u(s)|)$ over the interval $[0, t]$ is the same as $\gamma(\|u_{[0,t]}\|_\infty) = \gamma(\sup_{s \in [0,t]}(|u(s)|))$, because the function γ is increasing, so one may replace this term by $\gamma(\|u\|_\infty)$, where $\|u\|_\infty = \sup_{s \in [0,\infty)} \gamma(|u(s)|)$ is the sup norm of the input, because the solution $x(t)$ depends only on values $u(s), s \le t$ (so, one could equally well consider the input that has values $\equiv 0$ for all $s > t$).

Since, in general, $\max\{a, b\} \le a + b \le \max\{2a, 2b\}$, one can restate the ISS condition in a slightly different manner, namely, asking for the existence of some $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ (in general, different from the ones in the ISS definition) such that
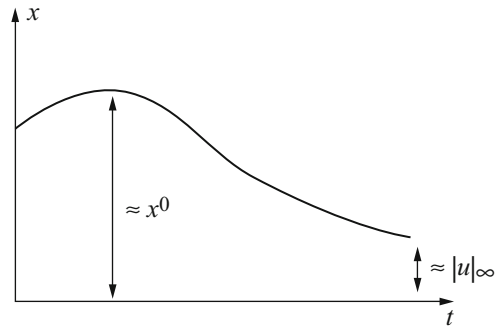
Input-to-State Stability, Fig. 1 ISS combines overshoot and asymptotic behavior

$$|x(t)| \le \max\left\{\beta(|x^0|, t), \ \gamma\left(\|u\|_\infty\right)\right\}$$

holds for all solutions. Such redefinitions, using "max" instead of sum, are also possible for each of the other concepts to be introduced later.

Intuitively, the definition of ISS requires that, for t large, the size of the state must be bounded by some function of the sup norm – that is to say, the amplitude – of inputs (because $\beta(|x^0|, t) \to 0$ as $t \to \infty$). On the other hand, the $\beta(|x^0|, 0)$ term may dominate for small t, and this serves to quantify the magnitude of the transient (overshoot) behavior as a function of the size of the initial state x^0 (Fig. 1). The *ISS superposition theorem*, discussed later, shows that ISS is, in a precise mathematical sense, the conjunction of two properties, one of them dealing with asymptotic bounds on $|x^0|$ as a function of the magnitude of the input and the other one providing a transient term obtained when one ignores inputs.

For internally stable linear systems $\dot{x} = Ax + Bu$, the variation of parameters formula gives immediately the following inequality:

$$|x(t)| \le \beta(t) |x^0| + \gamma \|u\|_\infty,$$

where

$$\beta(t) = \left\|e^{tA}\right\| \to 0 \quad \text{and}$$

$$\gamma = \|B\| \int_0^\infty \left\|e^{sA}\right\| ds < \infty.$$

This is a particular case of the ISS estimate, $|x(t)| \leq \beta(|x^0|, t) + \gamma(\|u\|_\infty)$, with linear comparison functions.

Feedback Redesign

The notion of ISS arose originally as a way to precisely formulate, and then answer, the following question. Suppose that, as in many problems in control theory, a system $\dot{x} = f(x, u)$ has been stabilized by means of a feedback law $u = k(x)$ (Fig. 2), that is to say, k was chosen such that the origin of the closed-loop system $\dot{x} = f(x, k(x))$ is globally asymptotically stable. (See, e.g., Sontag 1999 for a discussion of mathematical aspects of state feedback stabilization.) Typically, the design of k was performed by ignoring the effect of possible *input disturbances* $d(\cdot)$ (also called actuator disturbances). These "disturbances" might represent true noise or perhaps errors in the calculation of the value $k(x)$ by a physical controller or modeling uncertainty in the controller or the system itself. What is the effect of considering disturbances? In order to analyze the problem, d is incorporated into the model, and one studies the new system $\dot{x} = f(x, k(x) + d)$, where d is seen as an input (Fig. 3). One may then ask what is the effect of d on the behavior of the system. Disturbances d may well destabilize the system, and the problem may arise even when using a routine technique for control design, feedback linearization. To appreciate this issue, take the following very simple example. Given is the system

$$\dot{x} = f(x, u) = x + (x^2 + 1)u.$$

In order to stabilize it, substitute $u = \frac{\tilde{u}}{x^2+1}$ (a preliminary feedback transformation), rendering the system linear with respect to the new input $\tilde{u}$: $\dot{x} = x + \tilde{u}$, and then use $\tilde{u} = -2x$ in order to obtain the closed-loop system $\dot{x} = -x$. In other words, in terms of the original input u, the feedback law is

$$k(x) = \frac{-2x}{x^2 + 1}$$
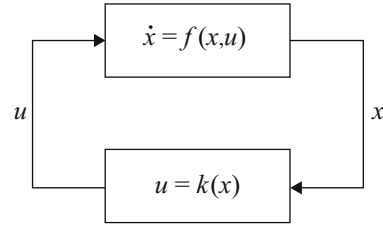
Input-to-State Stability, Fig. 2 Feedback stabilization, closed-loop system $\dot{x} = f(x, k(x))$
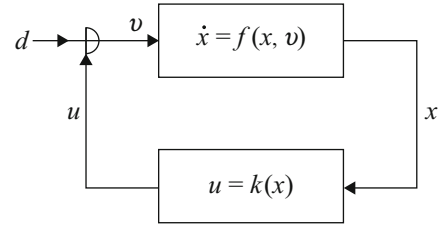
Input-to-State Stability, Fig. 3 Actuator disturbances, closed-loop system $\dot{x} = f(x, k(x) + d)$

so that $f(x, k(x)) = -x$. This is a GAS system. The effect of the disturbance input d is analyzed as follows. The system $\dot{x} = f(x, k(x) + d)$ is

$$\dot{x} = -x + (x^2 + 1)d.$$

This system has solutions which diverge to infinity even for inputs d that converge to zero; moreover, the constant input $d \equiv 1$ results in solutions that explode in finite time. Thus $k(x) = \frac{-2x}{x^2+1}$ was not a good feedback law, in the sense that its performance degraded drastically once actuator disturbances were taken into account.

The key observation for what follows is that if one adds a correction term "$-x$" to the above formula for $k(x)$, so that now,

$$\tilde{k}(x) = \frac{-2x}{x^2 + 1} - x,$$

then the system $\dot{x} = f(x, \tilde{k}(x) + d)$ with disturbance d as input becomes instead

$$\dot{x} = -2x - x^3 + (x^2 + 1)d$$

and this system is much better behaved: it is still GAS when there are no disturbances (it reduces

to $\dot{x} = -2x - x^3$), but, in addition, it is ISS (easy to verify directly, or appealing to some of the characterizations mentioned later). Intuitively, for large x, the term $-x^3$ serves to dominate the term $(x^2 + 1)d$, for all bounded disturbances $d(\cdot)$, and this prevents the state from getting too large.

This example is an instance of a general result, which says that, whenever there is some feedback law that stabilizes a system, there is also a (possibly different) feedback so that the system with external input d is ISS.

Theorem 1 (Sontag 1989). *Consider a system affine in controls*

$$\dot{x} = f(x, u) = g_0(x) + \sum_{i=1}^{m} u_i g_i(x) \ (g_0(0) = 0)$$

and suppose that there is some differentiable feedback law $u = k(x)$ so that

$$\dot{x} = f(x, k(x))$$

has $x = 0$ as a GAS equilibrium. Then, there is a feedback law $u = \widetilde{k}(x)$ such that

$$\dot{x} = f(x, \widetilde{k}(x) + d)$$

is ISS with input $d(\cdot)$.

The reader is referred to the book Krstić et al. (1995), and the references given later, for many further developments on the subjects of recursive feedback design, the "backstepping" approach, and other far-reaching extensions.

Equivalences for ISS

This section reviews results that show that ISS is equivalent to several other notions, including asymptotic gain, existence of robustness margins, dissipativity, and an energy-like stability estimate.

Nonlinear Superposition Principle

Clearly, if a system is ISS, then the system with no inputs $\dot{x} = f(x, 0)$ is GAS: the term $\|u\|_\infty$

vanishes, leaving precisely the GAS property. In particular, then, the system $\dot{x} = f(x, u)$ is *0-stable*, meaning that the origin of the system without inputs $\dot{x} = f(x, 0)$ is stable in the sense of Lyapunov: for each $\epsilon > 0$, there is some $\delta > 0$ such that $|x^0| < \delta$ implies $|x(t, x^0)| < \epsilon$. (In comparison-function language, one can restate 0-stability as follows: there is some $\gamma \in \mathcal{K}$ such that $|x(t, x^0)| \leq \gamma(|x^0|)$ holds for all small x^0.)

On the other hand, since $\beta(|x^0|, t) \to 0$ as $t \to \infty$, for t large one has that the first term in the ISS estimate $|x(t)| \leq \max\{\beta(|x^0|, t), \gamma(\|u\|_\infty)\}$ vanishes. Thus an ISS system satisfies the following *asymptotic gain property ("AG"):* there is some $\gamma \in \mathcal{K}_\infty$ so that:

$$\varlimsup_{t \to +\infty} |x(t, x^0, u)| \leq \gamma(\|u\|_\infty) \quad \forall x^0, u(\cdot) \tag{AG}$$

(see Fig. 4). In words, for all large enough t, the trajectory exists, and it gets arbitrarily close to a sphere whose radius is proportional, in a possibly nonlinear way quantified by the function γ, to the amplitude of the input. In the language of robust control, the estimate (AG) would be called an "ultimate boundedness" condition; it is a generalization of attractivity (all trajectories converge to zero, for a system $\dot{x} = f(x)$ with no inputs) to the case of systems with inputs; the "lim sup" is required since the limit of $x(t)$ as $t \to \infty$ may well not exist. From now on (and analogously when defining other properties), we
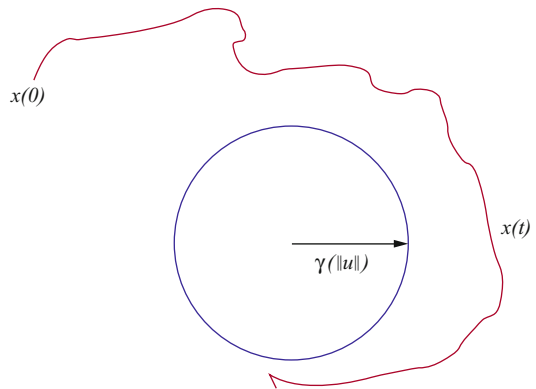
Input-to-State Stability, Fig. 4 Asymptotic gain property

will just say "the system is AG" instead of the more cumbersome "satisfies the AG property."

Observe that, since only large values of t matter in the limsup, one can equally well consider merely tails of the input u when computing its sup norm. In other words, one may replace $\gamma(\|u\|_\infty)$ by $\gamma(\overline{\lim}_{t \to +\infty} |u(t)|)$, or (since γ is increasing) $\overline{\lim}_{t \to +\infty} \gamma(|u(t)|)$.

The surprising fact is that these two necessary conditions are also sufficient. This is summarized by the *ISS superposition theorem*:

Theorem 2 (Sontag and Wang 1996). *A system is ISS if and only if it is 0-stable and AG.*

A minor variation of the above superposition theorem is as follows. Let us consider the *limit property (LIM)*:

$$\inf_{t \geq 0} |x(t, x^0, u)| \leq \gamma(\|u\|_\infty) \quad \forall\, x^0, u(\cdot) \quad \text{(LIM)}$$

(for some $\gamma \in \mathcal{K}_\infty$).

Theorem 3 (Sontag and Wang 1996). *A system is ISS if and only if it is 0-stable and LIM.*

Robust Stability

In this entry, a system is said to be *robustly stable* if it admits a *margin of stability* ρ, that is, a smooth function $\rho \in \mathcal{K}_\infty$ so the system

$$\dot{x} = g(x, d) := f(x, d\rho(|x|))$$

is GAS uniformly in this sense: for some $\beta \in \mathcal{KL}$,

$$\left| x(t, x^0, d) \right| \leq \beta(|x^0|, t)$$

for all possible $d(\cdot) : [0, \infty) \to [-1, 1]^m$. An alternative way to interpret this concept (cf. Sontag and Wang 1995) is as uniform global asymptotic stability of the origin with respect to all possible time-varying feedback laws Δ bounded by ρ: $|\Delta(t, x)| \leq \rho(|x|)$. In other words, the system

$$\dot{x} = f(x, \Delta(t, x))$$

(Fig. 5) is stable uniformly over all such perturbations Δ. In contrast to the ISS definition, which deals with all possible "open-loop" inputs, the
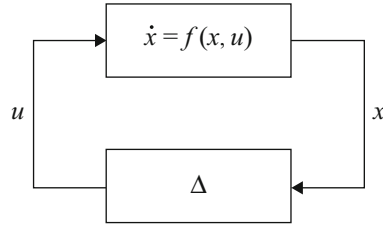
Input-to-State Stability, Fig. 5 Margin of robustness

present notion of robust stability asks about all possible closed-loop interconnections. One may think of Δ as representing uncertainty in the dynamics of the original system, for example.

Theorem 4 (Sontag and Wang 1995). *A system is ISS if and only if it is robustly stable.*

Intuitively, the ISS estimate $|x(t)| \leq \max\{\beta(|x^0|, t), \gamma(\|u\|_\infty)\}$ says that the β term dominates as long as $|u(t)| \ll |x(t)|$ for all t, but $|u(t)| \ll |x(t)|$ amounts to $u(t) = d(t).\rho(|x(t)|)$ with an appropriate function ρ. This is an instance of a "small gain" argument, see below. One analog for linear systems is as follows: if A is a Hurwitz matrix, then $A + Q$ is also Hurwitz, for all small enough perturbations Q; note that when Q is a nonsingular matrix, $|Qx|$ is a $\mathcal{K}_\infty$ function of $|x|$.

Dissipation

Another characterization of ISS is as a dissipation notion stated in terms of a Lyapunov-like function. A continuous function $V : \mathbb{R}^n \to \mathbb{R}$ is said to be a *storage function* if it is positive definite, that is, $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$, and proper, that is, $V(x) \to \infty$ as $|x| \to \infty$. This last property is equivalent to the requirement that the sets $V^{-1}([0, A])$ should be compact subsets of $\mathbb{R}^n$, for each $A > 0$, and in the engineering literature, it is usual to call such functions *radially unbounded*. It is an easy exercise to show that $V : \mathbb{R}^n \to \mathbb{R}$ is a storage function if and only if there exist $\underline{\alpha}, \overline{\alpha} \in \mathcal{K}_\infty$ such that

$$\underline{\alpha}(|x|) \leq V(x) \leq \overline{\alpha}(|x|) \quad \forall\, x \in \mathbb{R}^n$$

(the lower bound amounts to properness and $V(x) > 0$ for $x \neq 0$, while the upper bound guarantees $V(0) = 0$). For convenience, $\dot{V} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the function:

$$\dot{V}(x, u) := \nabla V(x) . f(x, u)$$

which provides, when evaluated at $(x(t), u(t))$, the derivative $dV(x(t))/dt$ along solutions of $\dot{x} = f(x, u)$.

An *ISS-Lyapunov function* for $\dot{x} = f(x, u)$ is by definition a smooth storage function V for which there exist functions $\gamma, \alpha \in \mathcal{K}_\infty$ so that

$$\dot{V}(x, u) \ \leq \ -\alpha(|x|) + \gamma(|u|) \quad \forall\, x, u\,.$$
(L-ISS)

Integrating, an equivalent statement is that, along all trajectories of the system, there holds the following dissipation inequality:

$$V(x(t_2)) - V(x(t_1)) \ \leq \ \int_{t_1}^{t_2} w(u(s), x(s))\, ds$$

where, using the terminology of Willems (1976), the "supply" function is $w(u, x) = \gamma(|u|) - \alpha(|x|)$. For systems with no inputs, an ISS-Lyapunov function is precisely the same object as a Lyapunov function in the usual sense.

Theorem 5 (Sontag and Wang 1995). *A system is ISS if and only if it admits a smooth ISS-Lyapunov function.*

Since $-\alpha(|x|) \leq -\alpha(\overline{\alpha}^{-1}(V(x)))$, the ISS-Lyapunov condition can be restated as

$$\dot{V}(x, u) \ \leq \ -\tilde{\alpha}(V(x)) + \gamma(|u|) \quad \forall\, x, u$$

for some $\tilde{\alpha} \in \mathcal{K}_\infty$. In fact, one may strengthen this a bit (Praly and Wang 1996): for any ISS system, there is a always a smooth ISS-Lyapunov function satisfying the "exponential" estimate $\dot{V}(x, u) \leq -V(x) + \gamma(|u|)$.

The sufficiency of the ISS-Lyapunov condition is easy to show and was already in the original paper Sontag (1989). A sketch of proof is as follows, assuming for simplicity a dissipation estimate in the form $\dot{V}(x, u) \leq -\alpha(V(x)) + \gamma(|u|)$. Given any x and u, either $\alpha(V(x)) \leq 2\gamma(|u|)$

or $\dot{V} \leq -\alpha(V)/2$. From here, one deduces by a comparison theorem that, along all solutions,

$$V(x(t)) \leq \max\left\{ \beta(V(x^0), t), \alpha^{-1}(2\gamma(\|u\|_\infty)) \right\},$$

where the $\mathcal{KL}$ function $\beta(s, t)$ is the solution $y(t)$ of the initial value problem

$$\dot{y} = -\frac{1}{2}\alpha(y) + \gamma(u), \quad y(0) = s.$$

Finally, an ISS estimate is obtained from $V(x^0) \leq \overline{\alpha}(x^0)$.

The proof of the converse part of the theorem is based upon first showing that ISS implies robust stability in the sense already discussed and then obtaining a converse Lyapunov theorem for robust stability for the system $\dot{x} = f(x, d\rho(|x|)) = g(x, d)$, which is asymptotically stable uniformly on all Lebesgue-measurable functions $d(\cdot) : \mathbb{R}_{\geq 0} \to B(0, 1)$. This last theorem was given in Lin et al. (1996) and is basically a theorem on Lyapunov functions for differential inclusions. The classical result of Massera (1956) for differential equations (with no inputs) becomes a special case.

Using "Energy" Estimates Instead of Amplitudes

In linear control theory, H_∞ theory studies $L^2 \to L^2$ induced norms, which under coordinate changes leads to the following type of estimate:

$$\int_0^t \alpha\,(|x(s)|))\, ds \ \leq \ \alpha_0(|x^0|) + \int_0^t \gamma(|u(s)|)\, ds$$

along all solutions and for some $\alpha, \alpha_0, \gamma \in \mathcal{K}_\infty$. Just for the statement of the next result, a system is said to *satisfy an integral-integral estimate* if for every initial state x^0 and input u, the solution $x(t, x^0, u)$ is defined for all $t > 0$ and an estimate as above holds. (In contrast to ISS, this definition explicitly demands that $t_{\max} = \infty$.)

Theorem 6 (Sontag 1998). *A system is ISS if and only if it satisfies an integral-integral estimate.*

This theorem is quite easy to prove, in view of previous results. A sketch of proof is as follows. If the system is ISS, then there is an ISS-Lyapunov function satisfying $\dot{V}(x, u) \le -V(x) + \gamma(|u|)$, so, integrating along any solution:

$$\int_0^t V(x(s))\,ds \le \int_0^t V(x(s))\,ds + V(x(t))$$
$$\le V(x(0)) + \int_0^t \gamma(|u(s)|)\,ds$$

and thus an integral-integral estimate holds. Conversely, if such an estimate holds, one can prove that $\dot{x} = f(x, 0)$ is stable and that an asymptotic gain exists.

Integral Input to State Stability

A concept of nonlinear stability that is truly distinct from ISS arises when considering a mixed notion which combines the "energy" of the input with the amplitude of the state. A system is said to be *integral-input to state stable (iISS)* provided that there exist $\alpha, \gamma \in \mathcal{K}_\infty$ and $\beta \in \mathcal{KL}$ such that the estimate

$$\alpha\left(|x(t)|\right) \le \beta(|x^0|, t) + \int_0^t \gamma(|u(s)|)\,ds \tag{iISS}$$

holds along all solutions. Just as with ISS, one could state this property merely for all times $t \in t_{\max}(x^0, u)$. Since the right-hand side is bounded on each interval $[0, t]$ (because, recall, inputs are by definition assumed to be bounded on each finite interval), it is automatically true that $t_{\max}(x^0, u) = +\infty$ if such an estimate holds along maximal solutions. So forward-completeness (solution exists for all $t > 0$) can be assumed with no loss of generality.

One might also consider the following type of "weak integral to integral" mixed estimate:

$$\int_0^t \underline{\alpha}(|x(s)|)\,ds \le \kappa(|x^0|)$$

$$+ \alpha\left(\int_0^t \gamma(|u(s)|)\,ds\right)$$

for appropriate $\mathcal{K}_\infty$ functions (note the additional "α").

Theorem 7 (Angeli et al. 2000b). *A system satisfies a weak integral to integral estimate if and only if it is iISS.*

Another interesting variant is found when considering mixed *integral/supremum* estimates:

$$\underline{\alpha}(|x(t)| \le \beta(|x^0|, t) + \int_0^t \gamma_1(|u(s)|)\,ds$$
$$+ \gamma_2(\|u\|_\infty)$$

for suitable $\beta \in \mathcal{KL}$ and $\underline{\alpha}, \gamma_i \in \mathcal{K}_\infty$. One then has

Theorem 8 (Angeli et al. 2000b). *A system satisfies a mixed estimate if and only if it is iISS.*

Dissipation Characterization of iISS
A smooth storage function V is an *iISS-Lyapunov function* for the system $\dot{x} = f(x, u)$ if there are a $\gamma \in \mathcal{K}_\infty$ and an $\alpha : [0, +\infty) \to [0, +\infty)$ which is merely *positive definite* (i.e., $\alpha(0) = 0$ and $\alpha(r) > 0$ for $r > 0$) such that the inequality

$$\dot{V}(x, u) \le -\alpha(|x|) + \gamma(|u|) \tag{L-iISS}$$

holds for all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$. To compare, recall that an ISS-Lyapunov function is required to satisfy an estimate of the same form but where α is required to be of class $\mathcal{K}_\infty$; since every $\mathcal{K}_\infty$ function is positive definite, an ISS-Lyapunov function is also an iISS-Lyapunov function.

Theorem 9 (Angeli et al. 2000a). *A system is iISS if and only if it admits a smooth iISS-Lyapunov function.*

Since an ISS-Lyapunov function is also an iISS one, ISS implies iISS. However, iISS is a strictly weaker property than ISS, because α may be bounded in the iISS-Lyapunov estimate, which means that V may increase, and the state become unbounded, even under bounded inputs, so long

as $\gamma(|u(t)|)$ is larger than the range of α. This is also clear from the iISS definition, since a constant input with $|u(t)| = r$ results in a term in the right-hand side that grows like rt.

An interesting general class of examples is given by *bilinear* systems

$$\dot{x} = \left(A + \sum_{i=1}^{m} u_i A_i \right) x + Bu$$

for which the matrix A is Hurwitz. Such systems are always iISS (see Sontag 1998), but they are not in general ISS. For instance, in the case when $B = 0$, boundedness of trajectories for all constant inputs already implies that $A + \sum_{i=1}^{m} u_i A_i$ must have all eigenvalues with nonpositive real part, for all $u \in \mathbb{R}^m$, which is a condition involving the matrices A_i (e.g., $\dot{x} = -x + ux$ is iISS but it is not ISS).

The notion of iISS is useful in situations where an appropriate notion of detectability can be verified using LaSalle-type arguments. There follow two examples of theorems along these lines.

Theorem 10 (Angeli et al. 2000a). *A system is iISS if and only if it is 0-GAS and there is a smooth storage function V such that, for some $\sigma \in \mathcal{K}_\infty$:*
$$\dot{V}(x, u) \leq \sigma(|u|)$$
for all (x, u).

The sufficiency part of this result follows from the observation that the 0-GAS property by itself already implies the existence of a smooth and positive definite, but not necessarily proper, function V_0 such that $\dot{V}_0 \leq \gamma_0(|u|) - \alpha_0(|x|)$ for all (x, u), for some $\gamma_0 \in \mathcal{K}_\infty$ and positive definite α_0 (if V_0 were proper, then it would be an iISS-Lyapunov function). Now, one uses $V_0 + V$ as an iISS-Lyapunov function (V provides properness).

Theorem 11 (Angeli et al. 2000a). *A system is iISS if and only if there exists an output function $y = h(x)$ (continuous and with $h(0) = 0$) which provides zero detectability ($u \equiv 0$ and $y \equiv 0 \Rightarrow x(t) \rightarrow 0$) and dissipativity in the*

following sense: there exists a storage function V and $\sigma \in \mathcal{K}_\infty$, α positive definite, so that

$$\dot{V}(x, u) \leq \sigma(|u|) - \alpha(h(x))$$

holds for all (x, u).

Angeli et al. (2000b) contains several additional characterizations of iISS.

Superposition Principles for iISS

There are also asymptotic gain characterizations for iISS. A system is *bounded energy weakly converging state (BEWCS)* if there exists some $\sigma \in \mathcal{K}_\infty$ so that the following implication holds:

$$\int_0^{+\infty} \sigma(|u(s)|) \, ds < +\infty \quad \Rightarrow$$
$$\liminf_{t \to +\infty} |x(t, x^0, u)| = 0 \quad \text{BEWCS}$$

(more precisely: if the integral is finite, then $t_{\max}(x^0, u) = +\infty$ and the liminf is zero). It is *bounded energy frequently bounded state (BEFBS)* if there exists some $\sigma \in \mathcal{K}_\infty$ so that the following implication holds:

$$\int_0^{+\infty} \sigma(|u(s)|) \, ds < +\infty \quad \Rightarrow$$
$$\liminf_{t \to +\infty} |x(t, x^0, u)| < +\infty \quad \text{BEFBS}$$

(again, meaning that $t_{\max}(x^0, u) = +\infty$ and the lim inf is finite).

Theorem 12 (Angeli et al. 2004). *The following three properties are equivalent for any given system $\dot{x} = f(x, u)$:*
- *The system is iISS.*
- *The system is BEWCS and 0-stable.*
- *The system is BEFBS and 0-GAS.*

Summary and Future Directions

This entry focuses on stability notions relative to steady states, but a more general theory is also

possible that allows consideration of more arbitrary attractors, as well as robust and/or adaptive concepts. Much else has been omitted from this entry. Most importantly, one of the key results is the *ISS small-gain theorem* due to Jiang et al. (1994), which provides a powerful sufficient condition for the interconnection of ISS systems being itself ISS.

Other topics not treated include, among many others, all notions involving outputs; ISS properties of time-varying (and in particular periodic) systems; ISS for discrete-time systems; questions of sampling, relating ISS properties of continuous and discrete-time systems; ISS with respect to a closed subset K; stochastic ISS; applications to tracking, vehicle formations ("leader to followers" stability); and averaging of ISS systems. Sontag (2006) may also be consulted for further references, a detailed development of some of these ideas, and citations to the literature for others. In addition, the textbooks Isidori (1999), Krstić et al. (1995), Khalil (1996), Sepulchre et al. (1997), Krstić and Deng (1998), Freeman and Kokotović (1996), and Isidori et al. (2003) contain many extensions of the theory as well as applications.

Cross-References

▶ Feedback Stabilization of Nonlinear Systems
▶ Fundamental Limitation of Feedback Control
▶ Linear State Feedback
▶ Lyapunov's Stability Theory
▶ Stability and Performance of Complex Systems Affected by Parametric Uncertainty
▶ Stability: Lyapunov, Linear Systems

Bibliography

Angeli D, Sontag ED, Wang Y (2000a) A characterization of integral input-to-state stability. IEEE Trans Autom Control 45(6):1082–1097

Angeli D, Sontag ED, Wang Y (2000b) Further equivalences and semiglobal versions of integral input to state stability. Dyn Control 10(2):127–149

Angeli D, Ingalls B, Sontag ED, Wang Y (2004) Separation principles for input-output and integral-input-to-state stability. SIAM J Control Optim 43(1):256–276

Freeman RA, Kokotović PV (1996) Robust nonlinear control design, state-space and Lyapunov techniques. Birkhauser, Boston

Isidori A (1999) Nonlinear control systems II. Springer, London

Isidori A, Marconi L, Serrani A (2003) Robust autonomous guidance: an internal model-based approach. Springer, London

Jiang Z-P, Teel A, Praly L (1994) Small-gain theorem for ISS systems and applications. Math Control Signals Syst 7:95–120

Khalil HK (1996) Nonlinear systems, 2nd edn. Prentice-Hall, Upper Saddle River

Krstić M, Deng H (1998) Stabilization of uncertain nonlinear systems. Springer, London

Krstić M, Kanellakopoulos I, Kokotović PV (1995) Nonlinear and adaptive control design. Wiley, New York

Lin Y, Sontag ED, Wang Y (1996) A smooth converse Lyapunov theorem for robust stability. SIAM J Control Optim 34(1):124–160

Massera JL (1956) Contributions to stability theory. Ann Math 64:182–206

Praly L, Wang Y (1996) Stabilization in spite of matched unmodelled dynamics and an equivalent definition of input-to-state stability. Math Control Signals Syst 9:1–33

Sepulchre R, Jankovic M, Kokotović PV (1997) Constructive nonlinear control. Springer, New York

Sontag ED (1989) Smooth stabilization implies coprime factorization. IEEE Trans Autom Control 34(4):435–443

Sontag ED (1998) Comments on integral variants of ISS. Syst Control Lett 34(1–2):93–100

Sontag ED (1999) Stability and stabilization: discontinuities and the effect of disturbances. In: Nonlinear analysis, differential equations and control (Montreal, 1998). NATO science series C, Mathematical and physical sciences, vol 528. Kluwer Academic, Dordrecht, pp 551–598

Sontag ED (2006) Input to state stability: basic concepts and results. In: Nistri P, Stefani G (eds) Nonlinear and optimal control theory. Springer, Berlin, pp 163–220

Sontag ED, Wang Y (1995) On characterizations of the input-to-state stability property. Syst Control Lett 24(5):351–359

Sontag ED, Wang Y (1996) New characterizations of input-to-state stability. IEEE Trans Autom Control 41(9):1283–1294

Willems JC (1976) Mechanisms for the stability and instability in feedback systems. Proc IEEE 64:24–35

Interactive Environments and Software Tools for CACSD

Vasile Sima
Advanced Research, National Institute for
Research & Development in Informatics,
Bucharest, Romania

Abstract

The main functional and support facilities offered by interactive environments and tools for computer-aided control system design (CACSD) and reference examples of such software systems are presented, from both a user and a developer perspective. The essential functions these environments should possess and requirements which should be satisfied are discussed. The importance of reliability and efficiency is highlighted, besides the desired friendliness and flexibility of the user interface. Widely used environments and software tools for CACSD, including MATLAB, Mathematica, Maple, and the SLICOT Library, serve as illustrative examples.

Keywords

Automatic control; Controller design; Numerical algorithms; Simulation; User interface

Introduction

The complexity of many processes or systems to be controlled, and the strong performance requirements to be fulfilled nowadays, makes it very difficult or even impossible to design suitable control laws and algorithms without resorting to computers and dedicated software tools. computer-aided control system design (CACSD) is the use of computer programs to support the creation, analysis, evaluation, or optimization of a control system design. CACSD is a specialization of computer-aided design (CAD) for control systems. CAD is used in many domains, to enhance designer's productivity and the design quality and to manage the design versions and documentation. CACSD is not a new paradigm, since the first such software systems have been developed about 50 years ago. See the historical overview in a companion paper.

The interactive environments and tools for CACSD have evolved significantly during the last decades, in parallel with the developments of numerical linear algebra, scientific computations, and computer hardware and software, including programming and networking capabilities. Starting from simple collections of specialized tools for solving well-defined system analysis and design problems, the CACSD became increasingly more sophisticated and powerful, allowing complicated tasks to be orchestrated for fully covering the stages of control engineering design, prototyping, and testing, including even the transfer to practical systems and applications. Modeling, system analysis and synthesis, and control system assessment are activities which are assisted by the nowadays advanced CACSD environments and software tools. The main aim is to help the designer to concentrate on the design problem itself, not on theoretical approaches, numerical algorithms, and computational details. Moreover, CACSD environments allow the developers and users to do conceptual thinking, but also programming and debugging at a higher level of abstraction, in comparison with standard programming languages, like Fortran, C/C++, or Java[TM].

There are both commercial or free and open-source CACSD environments and tools. State-of-the-art CACSD systems exist for several platforms (Windows, Linux/UNIX, and Mac OS X). Multiple high-speed CPUs, graphics cards, and large amounts of RAM are well suited to perform graphically and computationally intensive tasks. A common feature is the presence of a "friendly" graphical user interface, but often a dedicated command language is also available. The user interacts with the CACSD environment, e.g., by specifying the model or control structure, the design requirements, and the values of essential parameters or by selecting and combining

Interactive CACSD environment (e.g., MATLAB, Mathematica, Maple) (for modeling, simulation, analysis, synthesis, etc.)
Toolboxes or packages with executables or functions written in the environment language
(Graphical) User interface, Interactive language, Graphical functions, API
CACSD subroutine libraries (e.g., SLICOT)
Mathematical subroutine libraries (e.g., LAPACK, ARPACK, IMSL, NAG)
Computer-optimized mathematical libraries or their generators (e.g., BLAS, MKL, ATLAS)
Libraries of intrinsic functions (e.g., in Fortran or C/C++)

Interactive Environments and Software Tools for CACSD, Fig. 1 Hierarchy of the software components incorporated in an interactive CACSD environment

the tools to be used. The process can be repeated until a satisfactory behavior is obtained.

Usually, the underlying computational tools on which the interactive environments are based are hidden to the user. Moreover, software for extensive testing is not normally provided, but demonstrators running few examples are offered. Unfortunately, even mathematically simple problems of small dimension can conduct to wrong results when using unsuitable algorithms. Illustrative control-related examples are given, e.g., in Van Huffel et al. (2004). Since system analysis and design tasks usually involve sequential or iterative solution of large and complex subproblems, it follows that the quality of the intermediate results is of utmost importance. Consequently, the interactive environments for CACSD should be based on reliable, efficient, and thoroughly tested computational building blocks, which are called at the lower layers of calculations. These blocks constitute the computational engine of an interactive environment.

Figure 1 gives a typical hierarchy of the software components incorporated in an interactive CACSD environment.

Interactive Environments for CACSD

Main Functionality

A comprehensive set of functions of and requirements for interactive environments for control engineering are described in MacFarlane et al. (1989), but such a set has probably not yet been covered by any single environment. State-of-the-art interactive environments for CACSD include many attractive functional features:

- Define or find (via first principles or system identification) various system models (e.g., state-space models or transfer-function matrices) and convert between different representations
- Find reduced order (or simplified) models, which can more economically be used for simulation, control, prediction, etc.
- Analyze basic system properties, like stability, controllability, observability, stabilizability, detectability, minimality, properness, etc.
- Analyze interactively the behavior of a control system for various scenarios
- Provide alternative tools for different categories of users, from novice to expert, and from classical to "modern" or advanced analysis and synthesis techniques, in time domain or frequency domain
- Provide a wide range of tools, covering modeling, system identification, filtering, control system design, simulation, real-time behavior, hardware-in-the-loop simulation, and code generation for easy deployment and ensure their interoperability
- Allow the user to add extensions at various levels, new functions, interfaces, or even toolboxes or packages, which can be made available to a general community and allow customization

In addition to the functional and computational tools, essential components of an interactive environment are the user interface, the application program interface (API), and the support tools which enable to easily specify, document, and store a design solution, to visualize and interpret the results, to export them to other applications for further processing, to generate reports, etc.

A good paradigm for the data environment is object orientation.

It is a common feature of an interactive environment for CACSD to address the requirements of a large diversity of users, in various stages of familiarity with the environment. This feature is expressed, e.g., by the option to use either a graphical user interface or a command language to call and sequence various computational procedures. In addition, tools for easy building new computational or graphical procedures, or for managing the codes and results, are often included. The command language should operate both on low-level data constructs, such as a matrix, and on high-level ones (e.g., system objects), and it should allow operator overloading (e.g., taking $G_1 * G_2$ as the result of a series interconnection of the systems represented by the system objects G_1 and G_2).

An environment for CACSD should integrate advanced user interfaces and API, a collection of problem solvers based on reliable and efficient numerical and possibly symbolic algorithms, and tools for visualizing and interpreting the results. Widely used such environments are, for instance, MATLAB from The MathWorks, Inc., Mathematica from Wolfram Research, or Maple from Waterloo Maple Inc. (Maplesoft). Earlier developments of CACSD packages are surveyed in Frederick et al. (1991). There are also environments dedicated to modeling and simulation, which cover a broad range of technical and engineering computations, including those for mechanical, electrical, thermodynamic, hydraulic, pneumatic, or thermal systems. An example is Dymola, presented in a subsequent subsection.

Reference interactive environments and tools for CACSD are presented in the following (sub)sections.

Reference Interactive Environments

MATLAB (MATrix LABoratory) is an integrated, interactive environment for technical computing, visualization, and programming (MathWorks 2013). Based on a powerful high-level interpreter language and development tools, an easy-to-use, flexible, and customizable graphical user interface, complemented with attractive visualization capabilities, and open for extensions with new toolkits, MATLAB can be used for solving intricate scientific and engineering problems, as well as for the development and deployment of applications.

MATLAB$^{®}$ and Simulink$^{®}$ are registered trademarks of The MathWorks, Inc. MATLAB, Simulink, and several toolboxes, including System Identification Toolbox, Control System Toolbox, and Robust Control Toolbox, are suitable for solving various control engineering problems; other toolboxes, such as Signal Processing Toolbox, Optimization Toolbox, and Symbolic Math Toolbox, offer additional useful facilities. See http://www.mathworks.com/products/.

Simulink is a high-level implementation of the engineering approach, based on block diagrams, to analyze and design control systems. It is also a powerful modeling and multi-domain simulation and model-based design tool for dynamic systems, which supports hierarchical system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink offers a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. The models may include MATLAB algorithms, and the simulation results may be further processed to MATLAB. Managing projects (files, components, data), connecting to hardware for real-time testing, and deploying the designed system are additional, useful Simulink features. Real-Time Workshop code generation allows to speed up the design and implementation, by generating syntactically and semantically correct code which can be uploaded to the target machine.

MATLAB environment is very suitable for rapid prototyping, seen in a broad sense. This may include not only fully designing and implementing a new control law, testing it on a host computer, and deploying on a target computer but also support for developing and testing new mathematical or control theories and algorithms.

Born around 1980, MATLAB has evolved and improved impressively. Since 2004, two releases have been issued each year. There was a major change of the interface in Release 2012b, visible

both in the core MATLAB "Desktop" and in Simulink. The so-called Toolstrip interface replaces former menus and toolbars and includes tabs which group functionality for common tasks. A gallery of applications from the MATLAB family of products is additionally available and can be extended by the user.

MATLAB supports developing applications with graphical user interface (GUI) features; this itself can be done graphically using GUIDE (GUI development environment). MATLAB has support for object-oriented programming and interfacing with other languages or connecting to similar environments as Maple or Mathematica. When using the command-line interface, MATLAB helps the user, e.g., by showing the arguments of the typed MATLAB functions; also, MATLAB allows execution profiling, for increasing the computational efficiency, and its editor can suggest changes in the user functions (the so-called M-files) for improving the performance.

MATLAB users may upload their own contributions to the MATLAB Central website or may download tools developed by other people. User feedback is used by the MATLAB developers to improve the functionality, reliability, and efficiency of the computations.

Commercial competitors to MATLAB include Mathematica, Maple, and IDL; free open-source alternatives are, e.g., GNU Octave, FreeMat, and Scilab, intended to be mostly compatible with the MATLAB language. For instance, a set of free CACSD tools for GNU Octave version 3.6.0 or beyond has been very recently developed (see http://octave.sourceforge.net/control/). The Octave extension package called *control* is based on the SLICOT Library and includes functionalities for system identification, system analysis, control system design (including H_∞ synthesis), and model reduction, which are the basic steps of the control engineer design workflow.

Mathematica is an interactive environment which supports complete computational workflows, making it suitable for a convenient endeavor from ideas to deployed solutions (see http://www.wolfram.com/mathematica/).

Mathematica offers, e.g., tools for 2D and 3D data and function visualization and animation, numeric and symbolic tools for discrete and continuous calculus, a toolkit for adding user interfaces to applications, control systems libraries, tools for parallel programming, etc. High-performance computing capabilities include the use of packed and sparse arrays, multiple precision arithmetic, automatic multi-threading on multi-core computers (based on processor-specific optimized libraries), hardware accelerators, support for grid technology, and CUDA and OpenCL GPU hardware. Mathematica and SystemModeler (based on Modelica© language) offer numerous built-in functions which allow to design, analyze, and simulate continuous- and discrete-time control systems; simplify models; interactively test controllers; and document the design. Both classical and modern techniques are provided. A powerful symbolic-numeric computation engine and highly efficient numerical algorithms are used. Mathematica allows to define the system models in a more natural form than MATLAB. It can analyze not only numeric systems but also symbolic ones, represented by state-space or transfer-function models. The computational precision and algorithms can be automatically controlled and selected, respectively, and using arbitrary precision arithmetic is possible.

Maple is a computer algebra system, which combines a powerful engine for mathematical calculations with an intuitive user interface (see http://www.maplesoft.com/). Classical mathematical notation can be used, and the interface is customizable. Arbitrary precision numerical computations, as well as symbolic computations, can be performed. The Maple language is provided by a small kernel. NAG Numerical Libraries, ATLAS libraries, and other libraries written in this language are used for numerical calculations. Symbolic expressions are stored as directed acyclic graphs. The latest release, Maple 17, added hundreds of new problem-solving commands and interface enhancements. Many calculations recorded an impressive improvement in efficiency, compared

I

to the previous release. Examples include calculations with complex floating-point numbers and linear algebra operations. It is possible to use multiple cores and CPUs. The parallel memory management has been improved. Maple includes some CACSD tools for linear and nonlinear dynamic systems. For instance, the built-in package DynamicSystems (available since Maple 12 release) covers the analysis of linear time-invariant systems. Numerical solvers for Sylvester and Lyapunov equations have been added to the LinearAlgebra packages in Maple 13, and solvers for algebraic Riccati equations – based on SLICOT Library routines – have been included in Maple 14 (available in multiple precision arithmetic since Maple 15). Moreover, the MapleSim environment, based on Modelica, is dedicated to physical modeling and simulation. Symbolic simplification, numerical solution of the differential-algebraic equations (DAEs), and model post-processing (sensitivity analysis, linearization, parameter optimization, code generation, etc.) can be performed in MapleSim. Its Control Design Toolbox provides solutions for optimal control, Kalman filtering, pole assignment, etc. Bidirectional communication with MATLAB is possible.

MuPAD is another computer algebra system, initially developed by a group at the University of Paderborn, Germany, and then in cooperation with SciFace Software GmbH & Co. KG, company purchased in 2008 by The MathWorks, Inc. MuPAD has been used with Scilab, and now it is available in the Symbolic Math Toolbox. MuPAD is able to operate on formulas symbolically or numerically (with specified accuracy). It offers a programming language allowing object-oriented and functional programming, several packages for linear algebra, differential equations, number theory, and statistics, an interactive graphical system supporting animations and transparent areas in tridimensional images, etc.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench), from National Instruments, is an interactive development environment, based on MATRIXx, for a visual programming language mainly used for data

acquisition, instrument control, and industrial automation. Its Control Design and Simulation Module (see http://sine.ni.com/psp/app/doc/p/id/psp-648/lang/en) can be used to build process and controller models using transfer-function, state-space, or zero-pole-gain representations, analyze the open- and closed-loop system behavior, deploy the designed controllers to real-time hardware using built-in functions and LabVIEW Real-Time Module, etc.

Software Tools for CACSD

The software tools for CACSD are formally divided below into computational and support tools. SLICOT Library and Dymola serve as illustrative examples. The support tools can also include computational components.

Computational Tools

The computational tools for CACSD implement the main numerical algorithms of the systems and control theory and should satisfy several strong requirements:

- Reliability or guaranteed accuracy, which implies the use of numerically stable algorithms as much as possible and the estimation of the problem sensitivity (conditioning) and of the results accuracy; backward numerical stability ensures that the computed results are exact for slightly perturbed original data.
- Computational efficiency, which is important for large-scale engineering design problems or for real-time control.
- Robustness, which is mainly ensured by avoiding overflows, harmful underflows, and unacceptable accumulation of rounding-errors; scaling the data may be essential.
- Ease-of-use, achieved by simplified user interface (hiding the details), and default values for algorithmic parameters, such as tolerances.
- Wide scope and rich functionality, which address the range of problems and system representations that can be handled.
- Portability to various platforms, in the sense of functional correctness.
- Reusability, in building several dedicated engineering software systems or environments.

More details are given, e.g., in Van Huffel et al. (2004). An example addressing all these aspects is discussed in what follows.

SLICOT Library Benner et al. (1999) and Van Huffel et al. (2004) is one of the most comprehensive libraries for control theory numerical computations, containing over 500 subroutines which cover system analysis, benchmark and test problems, data analysis, filtering, identification, mathematical routines, some capabilities for nonlinear systems, synthesis, system transformation, and utility routines (see http://www.slicot. org/). The requirements above have been taken into account in the SLICOT Library development. Some of the SLICOT components are used in several interactive environments for CACSD, including MATLAB, Maple, Scilab, and Octave *control* package. The library is still under development. It is worth mentioning the new focus on structure-preserving algorithms, which offer increased accuracy, reliability, and efficiency, in comparison with standard solvers. Many procedures for optimal control and filtering, model reduction, etc., can benefit from using the "structured" solvers. There are also separate SLICOT-based toolboxes for MATLAB (Benner et al. 2010). SLICOT components follow predefined implementation and documentation standards.

SLICOT Library routines, and functions from many interactive environments for CACSD call components from the Basic Linear Algebra Subprograms (BLAS, see Dongarra et al. 1990 and the references therein) and Linear Algebra PACKage (LAPACK, Anderson et al. 1999). This approach enhances portability and efficiency, since optimized BLAS and LAPACK Libraries are provided for major computer platforms.

Support Tools

The support software tools for CACSD offer additional capabilities compared to computational tools. They may include alternative algorithms, symbolic computations (usually, for low-dimensional problems), and extended functionality, e.g., for modeling/simulation of nonlinear systems, code generation, etc. The support tools can be used by software developers of CACSD environments or computational tools

or directly by other users. For instance, symbolic calculations are useful for checking the accuracy of numerical algorithms. The code generation facility offers a safe and convenient support for deploying a design solution to the control hardware. A reference support software tool is briefly presented below.

Dymola (Dynamic modeling laboratory), from Dassault Systemes (see http://www.3ds.com/ products/catia/portfolio/dymola), deals with high-fidelity modeling and simulation of complex systems from various domains, like aerospace, automotive, robotics, process control, and other applications. Compatible and comprehensive model libraries, developed by leading experts, exist for many engineering branches. The users may create their own libraries or adapt existing libraries. This flexibility and openness is provided by the use of the open, object-oriented modeling language Modelica$^{©}$, currently further developed by the Modelica Association.

Equation-oriented models, based on DAEs, and symbolic manipulation are used, stimulating the reuse of components and enhancing the reliability and efficiency of the calculations. This approach enables to simplify generating the equations, which result from interconnecting various subsystems, and to deal with algebraic loops and structurally singular models. Algebraic loops are encountered when some auxiliary variables depend algebraically upon each other in a mutual way (Cellier and Elmqvist 1992). Structural singularities are related to DAE of index higher than 1.

Dymola allows performing hardware-in-the-loop simulation and real-time 3D animation. A model can be built by graphical composition, connecting components from various libraries using simple dragged-and-dropped operations. The parameters a model depends on can be tuned either by *parameter estimation* (also called *model calibration*), which minimizes the error between the physical measurements and simulation results, or by optimization, which minimizes certain performance criteria. Sometimes, e.g., when designing certain controllers, the criteria values are obtained by simulation. Dymola offers also facilities for model management, including

checking, testing, encrypting, or comparing models, and version control.

Summary and Future Directions

The main functional and support facilities offered by interactive environments and software tools for CACSD and reference examples have been presented. Their remarkable evolution during the past decades, combined with the importance of the design solutions they offer, is the strong argument that the CACSD software arsenal will continue to evolve and more reliable, efficient, and powerful systems will come into place. Progress is expected at all levels, including basic algorithms and numerical and symbolic libraries but also command languages, user interfaces, human-machine communication, and associated hardware. Tools for adaptive, nonlinear, and distributed control systems design should be developed and integrated. Artificial intelligence support might be required to add expert capabilities to the forthcoming interactive environments.

Cross-References

▶ Computer-Aided Control Systems Design: Introduction and Historical Overview
▶ Model Order Reduction: Techniques and Tools
▶ Multi-domain Modeling and Simulation
▶ Optimization-Based Control Design Techniques and Tools
▶ Robust Synthesis and Robustness Analysis Techniques and Tools
▶ System Identification: An Overview
▶ Validation and Verification Techniques and Tools

Recommended Reading

CACSD is well presented in many textbooks. A very recent one is Chin (2012), which covers modeling, control system design, implementation, and testing, and describes practical applications using MATLAB and Simulink. Many IFAC (International Federation of Automatic Control) and IEEE (Institute for Electrical and Electronics Engineers) international conferences and symposia have been dedicated to CACSD, going back more than two decades. A wealth of material is available, e.g., on IEEE Xplore (ieeexplore.ieee.org), containing the proceedings of many of the IEEE CACSD events. A recent event is the 2011 IEEE International Symposium on CACSD. Similar IEEE events were hold on 2010, 2008, 2006, 2004, 2002, 2000, 1999, 1996, 1994, 1992, and 1989. A new IEEE CACSD Conference, for Systems under Uncertainty, took place in July 2013.

Bibliography

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999) LAPACK users' guide, 3rd edn. SIAM, Philadelphia

Benner P, Mehrmann V, Sima V, Van Huffel S, Varga A (1999) SLICOT – a subroutine library in systems and control theory. In: Datta BN (ed) Applied and computational control, signals, and circuits, vol 1. Birkhäuser, Boston, pp 499–539

Benner P, Kressner D, Sima V, Varga A (2010) Die SLICOT-Toolboxen für Matlab. Automatisierungstechnik 58:15–25

Cellier FE, Elmqvist H (1992) The need for automated formula manipulation in object-oriented continuous-system modeling. In: Proceedings of the IEEE symposium on computer-aided control system design, Tucson, pp 1–8, 17–19 Mar 1992

Chin CS (2012) Computer-aided control systems design: practical applications using MATLAB® and Simulink®. CRC, Boca Raton

Dongarra JJ, Du Croz J, Duff IS, Hammarling S (1990) Algorithm 679: a set of level 3 basic linear algebra subprograms. ACM Trans Math Softw 16(1–17): 18–28

Frederick DK, Herget CJ, Kool R, Rimvall M (1991) ELCS – the extended list of control software. Report, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven

MacFarlane AGJ, Grübel G, Ackermann J (1989) Future design environments for control engineering. Automatica 25:165–176

MathWorks (2013) MATLAB® Primer. R2013a. The MathWorks, Inc., Natick

Van Huffel S, Sima V, Varga A, Hammarling S, Delebecque F (2004) High-performance numerical software for control. IEEE Control Syst Mag 24:60–76

Inventory Theory

Suresh P. Sethi
Jindal School of Management, The University of Texas at Dallas, Richardson, TX, USA

Abstract

This entry is a brief survey of classical inventory models and their extensions in several directions such as world-driven demands, presence of forecast updates, multi-delivery modes and advanced demand information, incomplete inventory information, and decentralized inventory control in the context of supply chain management. Important references are provided. We conclude with suggestions for future research.

Keywords

Base stock policy; EOQ model; Incomplete information; Newsvendor model; (s, S) policy

Introduction

Optimal inventory theory deals with managing stock levels of goods to effectively meet the demand of those goods. Because of the huge amount of capital that is tied up in inventory, its management is critical to the profitability of firms. A systematic analysis of inventory problems began with the development of the classical economic order quantity (EOQ) formula of Ford W. Harris in 1913. A substantial amount of research was reported in 1958 by Kenneth J. Arrow, Samuel Karlin, and Herbert Scarf, and much more has accumulated since then. Books on the topic include Zipkin (2000), Porteus (2002), Axsäter (2006), and Bensoussan (2011).

In this entry, we review single- and multi-period models with deterministic, stochastic, partially observed demand for a single product. In these models, our aim is to decide on the time of the orders and the order quantities. The time

between issuing an order and its receipt is called the lead time. For most of this review, we will assume the lead time to be zero, and the reader can consult the referenced books for nonzero lead time extensions and other topics not covered here.

Deterministic Demand

We will describe two classical models: the EOQ model and the dynamic lot size model.

The EOQ Model

This basic and most important deterministic model is concerned with a product that has a constant demand rate D in continuous time over an infinite horizon. No shortages are allowed. The costs consist of a fixed setup/ordering cost K and a holding cost h per unit of average on-hand stock per unit time. The production/purchase cost per unit time is a sunk cost since there is no choice of a total amount to produce, and hence it can be ignored. Although dynamic, the model can be reduced to a static model by a simple argument of periodicity. Moreover, it is obvious that one should never produce or order except for when the inventory level is zero, and one should order the same lot size Q each time the inventory level reaches zero. Since the average inventory level over time is $Q/2$ and the number of setups is D/Q per unit time, the long-run average cost to be minimized is $KD/Q+hQ/2$. The optimal policy that minimizes this cost, obtained using the first-order condition, is to order the lot size

$$Q = \sqrt{\frac{2KD}{h}} \qquad (1)$$

every time the inventory level reaches zero. Harris (1913) introduced the model. Erlenkotter (1990) provides a historical account of the formula, and Beyer and Sethi (1998) provide a mathematically rigorous proof involving quasi-variational inequalities (QVI) that arise in the course of dealing with continuous-time optimization problems involving fixed costs.

The Dynamic Lot Size Model

This is an analogue of the EOQ model when the demand varies over time. Wagner and Whitin (1958) developed it in the discrete-time finite horizon framework. With $D(t)$ denoting the demand in period t and other costs similar to those in the EOQ model, they showed that there exists an optimal policy in which an order will be issued just as the inventory level reaches zero, except for the first order. This policy is called the zero-inventory policy. With this in hand, the problem reduces to selecting only the order times. This is accomplished by applying a shortest path algorithm. Moreover, there are forward (recursion) procedures for solving the problem.

An important feature of this model is that in most cases, one can detect a *forecast horizon* which essentially separates earlier periods from later ones. More specifically, T is a forecast horizon if the first order in a T horizon problem remains optimal in any finite horizon problem with horizon longer than T, regardless of the demands beyond the period T. For an extensive bibliography of this literature, see Chand et al. (2002).

Stochastic Demand

We shall discuss three classical models and some of their extensions.

The Single-Period Problem: The Newsvendor Model

The problem of a newsvendor is to decide on an order quantity of newspapers to meet a stochastic demand at a minimum cost. If the realized demand is larger than the ordered quantity, it is lost and there is an opportunity loss of c_u (selling price minus purchase cost) for each paper short. On the other hand, for each paper ordered but not sold, there is an opportunity loss of c_o (purchase cost plus holding cost). The newsvendor conceptualizes the decision by each additional paper as a separate marginal contribution. The first is almost certain to be sold. Each additional paper is less likely to be sold than the previous one. Thus, each additional paper will be worth somewhat

less, and the marginal paper at the optimum should be worth exactly zero. Thus, c_u times the probability of selling the marginal paper minus c_o times the probability of not selling it should equal zero. Now, if F denotes the cumulative probability distribution function of the demand D, then clearly the optimal order quantity Q satisfies $c_o \cdot F(Q) - c_u \cdot (1 - F(Q)) = 0$, which gives us the famous newsvendor formula for the optimal order quantity

$$Q = F^{-1}\left(\frac{c_u}{c_u + c_o}\right), \qquad (2)$$

where $c_u / (c_u + c_o)$ is known as the critical fractile.

If p denotes the unit sale price, c the unit cost, and h the holding cost per unit per unit time, then $c_u = p - c$ and $c_o = c + h$, and therefore, the critical fractile can be expressed as $(p - c) / (p + h)$. An extension of the newsvendor formula to allow for a unit cost g of lost goodwill and a unit salvage value s received at the end of the period for each unit not sold is immediate. If we let $\alpha > 0$ denote the periodic discount factor, then $c_u = p + g - c$ and $c_o = c + h - \alpha s$ and the critical fractile becomes $(p + g - c) / (p + g + h - \alpha s)$, and therefore,

$$Q = F^{-1}\left(\frac{p + g - c}{p + g + h - \alpha s}\right). \qquad (3)$$

The newsvendor model has been used extensively in the context of supply chain management with multiple agents maximizing their individual objectives. In this case, inefficiencies arise due to double marginalization. Then, a question of appropriate contracts that can lead to the first-best solution, or coordinate the supply chain, becomes important. Cachon (2003) surveys this literature.

Multi-period Inventory Models: No Fixed Cost

The newsvendor model is a single-period model, and its multi-period generalization requires that the inventory not sold in a period is carried over to the next period. This results in the multi-period inventory model with lost sales. It is assumed

that demand in each period is independent and identically distributed (i.i.d.) with F denoting its cumulative probability distribution function. A rigorous analysis requires the method of dynamic programming, and it shows that there is a stock level S_t called *base stock* in period t, that we would ideally like to have at the beginning of period t. Thus, the optimal policy in period t, called *the base stock policy*, is to order

$$Q_t(x) = \begin{cases} S_t - x & \text{if } x < S_t, \\ 0 & \text{if } x \geq S_t. \end{cases} \tag{4}$$

In the special case when the terminal salvage value of an item is exactly equal to its cost c, it is possible to come up with the optimal policy using intuition. Since we do not need to salvage unused items in the multi-period setting, one can argue that an item carried over to the next period is worth its purchase cost c. Therefore, its presence means that the next period will need to order one less and thus save an amount c. In the last period, when there is no next period, our terminal salvage value assumption also guarantees a leftover item's worth to be also c. Thus, we can modify (3) and obtain a stationary base stock level

$$S = F^{-1} \left(\frac{p + g - c}{(p + g - c) + (c + h - \alpha c)} \right)$$
$$= F^{-1} \left(\frac{p + g - c}{p + g + h - \alpha c} \right) \tag{5}$$

for each period t.

Thus, the elimination of the endgame effect delivers us a *myopic policy*, a policy optimal in the single-period case to be also optimal in the dynamic multi-period setting. A more general concept than the optimality of a myopic policy is that of the forecast horizon mentioned earlier in the context of the dynamic lot size model.

Sometimes, when the demand exceeds the on-hand inventory in the period, the demand is not lost but backlogged. In this case, each unit of backlogged demand is satisfied in the next period, and unit revenue p is recovered, but a unit backlogging cost b is incurred, due to expediting, special handling, delayed receipt of revenue, and loss of goodwill. Thus, $c_u = b - (1 - \alpha)c$,

where the second term represents the savings due to postponing the purchase of the backlogged demand unit by one period, and $c_o = c + h - \alpha c$ as in (4). This gives us the base stock level

$$S = F^{-1} \left(\frac{b - (1 - \alpha)c}{b + h} \right), \tag{6}$$

which can be used in (5) to give the optimal policy.

Sometimes it is possible to have multiple delivery modes such as fast, regular, and slow as well as demand forecast updates. Then, at the beginning of each period, on-hand inventory and demand information are updated. At the same time, decisions on how much to order using each of the modes are made. Fast, regular, and slow orders are delivered at the ends of the current, the next, and one beyond the next periods, respectively. In such models, a modified base stock policy is optimal only for the two fastest modes. For details and further generalization, see Sethi et al. (2005).

An important extension includes serial inventory systems where stage 1 receives supplies from an outside source and each downstream stage receives supplies from its immediate upstream stage. Clark and Scarf (1960) introduced the notion of the echelon inventory position at a stage to consist of the stock at that stage plus stock in transit to that stage plus all downstream stock minus the amount backlogged at the final stage. Then, the optimal ordering policy at each stage is given by an echelon base stock policy with respect to the echelon inventory position at that stage. It is known that assembly systems can be reduced to a serial system. Details can be found in Zipkin (2000).

Multi-period Inventory Models: Fixed Cost

When there is a fixed cost of ordering, it is clear that it would not be reasonable to follow the base stock policy when the inventory level is not much below the base stock level. Indeed, Scarf (1960) proved that there are numbers s_t and S_t, $s_t < S_t$, for period t such that the optimal policy in period t is to order

$$Q_t(x) = \begin{cases} S_t - x & \text{if } x \leq s_t \\ 0 & \text{if } x > s_t. \end{cases} \tag{7}$$

Such a policy is famously known as an (s, S) policy.

When the demands are not i.i.d., the model has been extended to Markovian demands. In this case, there is an exogenous Markov process, and the distribution of the demand in each period depends on the state of the Markov process, called the demand state, in that period. It can be shown that the optimal policy in period t is (s_t^i, S_t^i), where i denotes the demand state in the period. Such a policy is also called a state-dependent (s, S) policy. Further details are available in Beyer et al. (2010). Recent advances in information technology have allowed managers to obtain advance demand information in addition to forecast updates. In such cases, a state-dependent (s, S) policy can be shown to be optimal. For details, refer to Ozer (2011).

The Continuous-Time Model: Fixed Cost

The marriage of the two classical results (1) and (7) is accomplished by Presman and Sethi (2006) in a continuous-time stochastic inventory model involving a demand that is the sum of a constant demand rate and a compound Poisson process. The optimal policies that minimize a discounted cost or the long-run average cost are both of (s, S) type. The (s, S) policy minimizing the long-run average cost reduces to the EOQ formula when the intensity of the compound Poisson process is set to zero. And when the constant demand component vanishes, the model reduces to the continuous-review stochastic inventory model with fixed cost and compound Poisson demand.

Incomplete Inventory Information Models (i3)

A critical assumption in the vast inventory theory literature has been that the level of inventory at any given time is fully observed. The celebrated results (1) and (7) have been obtained under the assumption of full observation. Yet the inventory level is often not fully observed in practice, for a variety of reasons such as replenishment errors, employee theft, customer shoplifting, improper handling and damaging of merchandise, misplaced inventories, uncertain yield, imperfect inventory audits, and incorrect recording of sales. In such an environment of incomplete information, inventories are known to be partially observed and most of the well-known inventory policies including (1) and (7) are not even admissible, let alone optimal. In such cases, Bensoussan et al. (2010) show that the dynamic programming equation can be written in terms of the unnormalized conditional probability of the current inventory level given past observations, referred to as signals, instead of just the inventory level in the full observation case. Furthermore, one can write the evolution of the conditional probability in terms of its current value, the current order, and the current observation. However, there are no longer simple optimal policies except in cases of information delay reported in Bensoussan et al. (2009) where modified base stock and (s, S) policies are shown to be optimal.

Summary and Future Directions

We briefly describe some classical results in inventory theory. These are based on full observation. Some recent work on inventory models under incomplete information is reported. This work leads to a number of new research directions, both theoretical and empirical as reported in Sethi (2010). It would be of much interest to know the industries where the i3 problem is serious enough to warrant the difficult mathematical analysis required. Furthermore, how are the observed signals related to the inventory level? It is also clear from the reviewed literature that there are no simple optimal policies for most i3 problems, so it would be important to develop efficient computational procedures to obtain optimal solutions or to specify a class of simple implementable policies and optimize within this class. An important benefit of solving i3 problems optimally is the provision of an economic justification for technologies such as RFID that may reduce inaccuracies in inventory observations.

Another area of research would be to study multi-period multi-agent supply chains with a stochastic inventory dynamics. While these can be formulated as dynamic games, there are a number of equilibrium concepts to deal with,

depending on the information the agents have. Some of them are time consistent or subgame perfect and some are not. Regardless, there are inefficiencies that arise from these decentralized game settings, and developing contracts for coordinating dynamic supply chains remains a wide open topic of research.

Cross-References

Bibliography

Arrow KJ, Karlin S, Scarf H (1958) Studies in the mathematical theory of inventory and production. Stanford University Press, Stanford

Axsäter S (2006) Inventory control. Springer, New York

Bensoussan A (2011) Dynamic programming and inventory control. IOS Press, Washington, DC

Bensoussan A, Cakanyildirim M, Feng Q, Sethi SP (2009) Optimal ordering policies for stochastic inventory problems with observed information delays. Prod Oper Manag 18(5):546–559

Bensoussan A, Cakanyildirim M, Sethi SP (2010) Filtering for discrete-time Markov processes and applications to inventory control with incomplete information. In: Crisan D, Rozovsky B (eds) Handbook on nonlinear filtering. Oxford University Press, Oxford, UK, pp 500–525

Beyer D, Cheng F, Sethi SP, Taksar MI (2010) Markovian demand inventory models. International series in operations research and management science. Springer, New York

Beyer D, Sethi SP (1998) A proof of the EOQ formula using quasi-variational inequalities. Int J Syst Sci 29(11):1295–1299

Cachon GP (2003) Supply chain coordination with contracts. In: de Kok AG, Graves S (eds) Supply chain management-handbook in OR/MS, chapter 6. North-Holland, Amsterdam, pp 229–340

Chand S, Hsu VN, Sethi SP (2002) Forecast, solution and rolling horizons in operations management problems: a classified bibliography. Manuf Serv Oper Manag 4(1):25–43

Clark AJ, Scarf H (1960) Optimal policies for a multi-echelon inventory problem. Manag Sci 6(4):475–490

Erlenkotter D (1990) Ford Whitman Harris and the economic order quantity model. Oper Res 38(6):937–946

Harris FW (1913) How many parts to make at once. Fact Mag Manag 10(2):135–136, 152. Reprinted (1990) Oper Res 38(6):947–950

Ozer O (2011) Inventory management: information, coordination, and rationality. In: Kempf KG, Keskinocak P, Uzsoy R (eds) Planning production and inventories in the extended enterprise. Springer, New York

Porteus EL (2002) Stochastic inventory theory. Stanford University Press, Stanford, CA

Presman E, Sethi SP (2006) Inventory models with continuous and Poisson demands and discounted and average costs. Prod Oper Manag 15(2):279–293

Scarf H (1960) The optimality of (s, S) policies in dynamic inventory problems. In: Arrow KJ, Karlin S, Suppes P (eds) Mathematical methods in the social sciences. Stanford University Press, Stanford, pp 196–202

Sethi SP (2010) i3: incomplete information inventory models. Decis Line Oct:16–19

Sethi SP, Yan H, Zhang H (2005) Inventory and supply chain management with forecast updates. Springer, New York

Wagner HM, Whitin TM (1958) Dynamic version of the economic lot size model. Manag Sci 5:89–96

Zipkin P (2000) Foundations of inventory management. McGraw Hill, New York

Investment-Consumption Modeling

L.C.G. Rogers
University of Cambridge, Cambridge, UK

Abstract

The simplest investment-consumption problem is the celebrated example of Robert Merton (J Econ Theory 3(4):373-413, 1971). This survey shows three different ways of solving the problem, each of which is a valuable solution method for more complicated versions of the question.

Keywords

Budget constraint; Hamilton-Jacobi-Bellman (HJB) equation; Merton problem; Value function

Introduction

Consider an investor in a market with a riskless bank account accruing continuously compounded

interest at rate r_t, and with a single risky asset whose price S_t at time t evolves as

$$dS_t = S_t(\sigma_t dW_t + \mu_t dt), \qquad (1)$$

where W is a standard Brownian motion, and σ and μ are processes previsible with respect to the filtration of W. The investor starts with initial wealth w_0 and chooses the rate c_t of consuming, and the wealth θ_t to invest in the risky asset, so that his overall wealth evolves as

$$dw_t = \theta_t(\sigma_t dW_t + \mu_t dt) + r_t(w_t - \theta_t)dt - c_t dt \tag{2}$$

$$= r_t w_t\, dt + \theta_t\{\sigma_t dW_t + (\mu_t - r_t)dt\} - c_t\, dt. \tag{3}$$

For convenience, we assume that σ, σ^{-1}, and μ are bounded. See Rogers and Williams (2000a,b) for background information on stochastic processes. The three terms in (2) have natural interpretations: The first expresses the evolution of the wealth invested in the stock, the second the interest accruing on the wealth $(w - \theta)$ invested in the bank account, and the third is the cash being withdrawn for consumption.

To avoid so-called doubling strategies, we insist that the wealth process so generated by the controls (c, θ) must remain bounded below in some suitable way, which here is just the condition $w_t \geq 0$ for all $t \geq 0$; any (c, θ) satisfying this condition will be called *admissible*. The set of admissible (c, θ) will be denoted $\mathcal{A}(w_0)$, a notation which makes explicit the dependence on the investor's initial wealth.

The investor's objective is taken to be to obtain

$$V(w_0) \equiv \sup_{(c,\theta)\in\mathcal{A}(w_0)} E\left[\int_0^\infty e^{-\rho t} U(c_t)\, dt\right] \tag{4}$$

for some constant $\rho > 0$. The problem cannot be solved explicitly at this level of generality, but if we take some special cases, we are able to illustrate the main methods used to attack it. Many other objectives with various different constraints can be handled by similar techniques: see Rogers (2013) for a wide range of examples.

The Main Techniques

We present here three important techniques for solving such problems: the value function approach; the use of dual variables; and the use of martingale representation. The first two methods only work if the problem is Markovian; the third only works if the market is complete. There is a further method, the Pontryagin-Lagrange approach; see Sect. 1.5 in Rogers (2013). While this is a quite general approach, we can only expect explicit solutions when further structure is available.

The Value Function Approach

To illustrate this, we focus on the original Merton problem (Merton 1971), where σ and μ are both constant, and the utility U is constant relative risk aversion (CRRA):

$$U'(x) = x^{-R} \qquad (x > 0) \tag{5}$$

for some $R > 0$ different from 1. The case $R = 1$ corresponds to logarithmic utility, and can be solved by similar methods. Perhaps the best starting point is the *Davis-Varaiya Martingale Principle of Optimal Control (MPOC)*: The process $Y_t = e^{-\rho t} V(w_t) + \int_0^t e^{-\rho s} U(c_s)\, ds$ is a supermartingale under any control, and a martingale under optimal control. If we use Itô's formula, we find that

$$e^{\rho t} dY_t = -\rho V(w_t)dt + V'(w_t)dw_t$$
$$+ \tfrac{1}{2}\sigma^2\theta_t^2 V''(w_t)dt + U(c_t)dt$$
$$\doteq [-\rho V + \{\theta_t(\mu - r) - c_t + r\}V'$$
$$+ \tfrac{1}{2}\sigma^2\theta_t^2 V'' + U(c_t)]\, dt, \tag{6}$$

where the symbol $\doteq$ denotes that the two sides differ by a (local) martingale. If the MPOC is to hold, then we expect that the drift in dY should be non-positive under any control, and equal to zero under optimal control. We simply assume for now that local martingales are martingales; this is of course not true in general, and is a point that needs to be handled carefully in a rigorous proof. Directly from (6), we then deduce the

Hamilton-Jacobi-Bellman (HJB) equations for this problem:

$$0 = \sup_{c,\theta}[-\rho V + \{\theta(\mu - r) - c + r\}V' + \tfrac{1}{2}\sigma^2\theta^2 V'' + U(c)]. \quad (7)$$

Write $\tilde{U}(y) \equiv \sup \{U(x) - xy\}$ for the convex dual of U, which in this case has the explicit form

$$\tilde{U}(y) = -\frac{y^{1-R'}}{1-R'} \quad (8)$$

with $R' \equiv 1/R$. We are then able to perform the optimizations in (7) quite explicitly to obtain

$$0 = -\rho V + rV' + \tilde{U}(V') - \tfrac{1}{2}\kappa^2 \frac{(V')^2}{V''} \quad (9)$$

where

$$\kappa \equiv \frac{\mu - r}{\sigma}. \quad (10)$$

Nonlinear PDEs arising from stochastic optimal control problems are not in general easy to solve, but (9) is tractable in this special setting, because the assumed CRRA form of U allows us to deduce by a scaling argument that $V(w) \propto w^{1-R} \propto U(w)$, and we find that

$$V(w) = \gamma_M^{-R} U(w), \quad (11)$$

where

$$R\gamma_M = \rho + (R-1)(r + \tfrac{1}{2}\kappa^2/R). \quad (12)$$

The optimal investment and consumption behavior is easily deduced from the optimal choices which took us from (7) to (9). After some calculations, we discover that

$$\theta_t^* = \pi_M w_t \equiv \frac{\mu - r}{\sigma^2 R} w_t, \qquad c_t^* = \gamma_M w_t \quad (13)$$

specifies the optimal investment/consumption behavior in this example. (The positivity of γ_M is necessary and sufficient for the problem to be well posed; see Sect. 1.6 in Rogers (2013)). Unsurprisingly, the optimal solution scales linearly with wealth.

Dual Variables

We illustrate the use of dual variables in the constant-coefficient case of the previous section, except that we no longer suppose the special form (5) for U. The analysis runs as before all the way to (9), but now the convex dual $\tilde{U}$ is not simply given by (8). Although it is not now possible to guess and verify, there is a simple transformation which reduces the nonlinear ODE (9) to something we can easily handle. We introduce the new variable $z > 0$ related to w by $z = V'(w)$, and define a function J by

$$J(z) = V(w) - wz. \quad (14)$$

Simple calculus gives us $J' = -w$, $J'' = -1/V''$, so that the HJB equation (9) transforms into

$$0 = \tilde{U}(z) - \rho J(z) + (\rho - r)zJ'(z) + \tfrac{1}{2}\kappa^2 z^2 J''(z), \quad (15)$$

which is now a second-order *linear* ODE, which can be solved by traditional methods; see Sect. 1.3 of Rogers (2013) for more details.

Use of Martingale Representation

This time, we shall suppose that the coefficients μ_t, r_t, and σ_t in the wealth evolution (3) are general previsible processes; to keep things simpler, we shall suppose that μ, r, and σ^{-1} are all bounded previsible processes. The Markovian nature of the problem which allowed us to find the HJB equation in the first two cases is now destroyed, and a completely different method is needed. The way in is to define a positive semimartingale ζ by

$$d\zeta_t = \zeta_t(-r_t dt - \kappa_t dW_t), \qquad \zeta_0 = 1 \quad (16)$$

where $\kappa_t = (\mu_t - r_t)/\sigma_t$ is a previsible process, bounded by hypothesis. This process, called the *state-price density process*, or the *pricing kernel*, has the property that if w evolves as (3), then $M_t \equiv \zeta_t w_t + \int_0^t \zeta_s c_s \, ds$ is a positive local martingale.

Since positive local martingales are super-martingales, we deduce from this that

$$M_0 = w_0 \geq E\left[\int_0^\infty \zeta_s c_s \, ds\right]. \qquad (17)$$

Thus, for any $(c, \theta) \in \mathcal{A}(w_0)$, the budget constraint (17) must hold. So the solution method here is to maximize the objective (4) subject to the constraint (17). Absorbing the constraint with a Lagrange multiplier λ, we find the unconstrained optimization problem

$$\sup E\left[\int_0^\infty \{e^{-\rho s} U(c_s) - \lambda \zeta_s c_s\} \, ds\right] + \lambda w_0 \qquad (18)$$

whose optimal solution is given by

$$e^{-\rho s} U'(c_s) = \lambda \zeta_s, \qquad (19)$$

and this determines the optimal c, up to knowledge of the Lagrange multiplier λ, whose value is fixed by matching the budget constraint (17) with equality.

Of course, the missing logical piece of this argument is that if we are given some $c \geq 0$ satisfying the budget constraint, is there necessarily some θ such that the pair (c, θ) is admissible for initial wealth w_0? In this setting, this can be shown to follow from the Brownian integral representation theorem, since we are in a complete market; however, in a multidimensional setting, this can fail, and then the problem is effectively insoluble.

Summary and Future Directions

This brief survey states some of the main ideas of consumption-investment optimization, and sketches some of the methods in common use. Explicit solutions are rare, and much of the interest of the subject focuses on efficient numerical schemes, particularly when the dimension of the problem is large. A further area of interest is in continuous-time principal-agent problems; Cvitanic and Zhang (2012) is a recent account of some of the methods of this subject, but it has to be said that the theory of such problems is much less complete than the simple single-agent optimization problems discussed here.

Cross-References

▶ Risk-Sensitive Stochastic Control
▶ Stochastic Dynamic Programming
▶ Stochastic Linear-Quadratic Control
▶ Stochastic Maximum Principle

Bibliography

Cvitanic J, Zhang J (2012) Contract Theory in Continuous-time Models. Springer, Berlin/Heidelberg

Merton RC (1971) Optimum consumption and portfolio rules in a continuous-time model. J Econ Theory 3(4):373–413

Rogers LCG (2013) Optimal Investment. Springer, Berlin/New York

Rogers LCG, Williams D (2000a) Diffusions, Markov Processes and Martingales, vol 1. Cambridge University Press, Cambridge/New York

Rogers LCG, Williams D (2000b) Diffusions, Markov Processes and Martingales, vol 2. Cambridge University Press, Cambridge

ISS

▶ Input-to-State Stability

Iterative Learning Control

David H. Owens
University of Sheffield, Sheffield, UK

Synonyms

ILC

Abstract

Iterative learning control addresses tracking control where the repetition of a task allows improved tracking accuracy from task to task. The area inherits the analysis and design issues of classical control but adds convergence conditions

for task to task learning, the need for acceptable task-to-task performance and the implications of modeling errors for task-to-task robustness.

Keywords

Adaptation; Optimization; Repetition; Robustness

Introduction

Iterative learning control (ILC) is relevant to trajectory tracking control problems on a finite interval $[0, T]$ (Ahn et al. 2007b; Bien and Xu 1998; Chen and Wen 1999). It has close links to multipass process theory (Edwards and Owens 1982) and repetitive control (Rogers et al. 2007) plus conceptual links to adaptive control. It focuses on problems where the repetition of a specified task creates the possibility of improving tracking accuracy from task to task and, in principle, reducing the tracking error to exactly zero. The iterative nature of the control schemes proposed, the use of past executions of the control to update/improve control action, and the asymptotic learning of the required control signals put the topic in the area of adaptive control, although other areas of study are reflected in its methodologies.

Application areas include robotic assembly (Arimoto et al. 1984), electromechanical test systems (Daley et al. 2007), and medical rehabilitation robotics (Rogers et al. 2010). For example, consider a manufacturing robot required to undertake an indefinite number of identical tasks (such as "pick and place" of components) specified by a spatial trajectory on a defined time interval. The problem is two-dimensional. More precisely, the controlled system evolves with *two* variables, namely, time $t \in [0, T]$ (elapsed in each iteration) and iteration index $k \geq 0$. Data consists of signals $f_k(t)$ denoting the value of the signal f at time t on iteration k. The conceptual algorithm used is:

Step one: (Preconditioning) Implement loop controllers to condition plant dynamics.

Step two: (Initialization) Given a demand signal $r(t)$, $t \in [0, T]$, choose an initial input $u_0(t)$, $t \in [0, T]$ and set $k = 0$.

Step three: (Response measurement) Return the plant to a defined initial state. Find the output response y_k to the input u_k. Construct the tracking error $e_k = r - y_k$. Store data.

Step four: (Input signal update) Use past records of inputs used and tracking errors generated to construct a new input $u_{k+1}(t)$, $t \in [0, T]$ to be used to improve tracking accuracy on the next trial.

Step five: (Termination/task repetition) Either terminate the sequence or increase k by unity and return to step 3.

It is the updating of the input signal based on observation that provides the conceptual link to adaptive control. ILC *causality* defines *"past data"* at time t on iteration k as data on the interval $[0, t]$ on that iteration plus all data on $[0, T]$ on all previous iterations. Feedback plus feedforward control normally contains feedforward transfer of information from past iterations to the current iteration.

Modeling Issues

Design approaches have been model-based. Most nonlinear problems assume nonlinear state space models relating the $\ell \times 1$ input vector $u(t)$ to the $m \times 1$ output vector $y(t)$ via an $n \times 1$ state vector $x(t)$ as follows:

$$\dot{x}(t) = f(x(t), u(t)), \quad y(t) = h(x(t), u(t)),$$

where $t \in [0, T]$, $x(0) = x_0$ and f and h are vector-valued functions. The discrete time (sample data) version replaces derivatives by a forward shift, where t is now a sample counter, $0 \leq t \leq N$ (the index of the last sample). The continuous time linear model is

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with an analogous model for discrete systems. In both cases, the matrices A, B, C, D are constant or time varying of appropriate dimension.

Nonlinear systems present the greatest technical challenge. Linear system's challenges are greater for the time-varying, continuous time case. The simplest linear case of discrete time, time-invariant systems can be described by a matrix relationship

$$y = Gu + d \qquad (1)$$

where y denotes the $m(N + 1) \times 1$ "supervector" generated by the time series $y(0), y(1), \ldots, y(N)$ and the construction $y = \left[y^T(0), y^T(1), \ldots, y^T(N) \right]^T$, the supervector u is generated, similarly, by the time series $u(0), u(1), \ldots, u(N)$, and d is generated by the times series $Cx_0, CAx_0, \ldots, CA^N x_0$. The matrix G has the lower block triangular structure

$$G = \begin{bmatrix} D & 0 & 0 \cdots 0 \\ CB & D & 0 \cdots 0 \\ CAB & CB & D \cdots 0 \\ \vdots & & \\ CA^{N-1}B & CA^{N-2}B & \cdots D \end{bmatrix}$$

defined in terms of the Markov parameter matrices $D, CB, CAB, \ldots$ of the plant. This structure has led to a focus on the discrete time, time-invariant case, and exploitation of matrix algebra techniques.

More generally, $G : \mathcal{U} \to \mathcal{Y}$ can be a bounded linear operator between suitable signal spaces $\mathcal{U}$ and $\mathcal{Y}$. Taking G as a convolution operator, the representation (1) also applies to time-varying continuous time and discrete time systems. The representation also applies to differential-delay systems, coupled algebraic and differential systems, multi-rate systems, and other situations of interest.

Formal Design Objectives

Problem Statement: *Given a reference signal $r \in \mathcal{Y}$ and an initial input signal $u_0 \in \mathcal{U}$, construct a causal control update rule/algorithm*

$$u_{k+1} = \psi_k(e_{k+1}, e_k, \ldots, e_0, u_k, u_{k-1}, \ldots, u_0)$$

that ensures that $\lim_{k \to \infty} e_k = 0$ (convergence) in the norm topology of $\mathcal{Y}$.

The update rule $\psi_k(\cdot)$ represents the simple idea of expressing u_{k+1} in terms of past data. A general linear "high-order" rule is

$$u_{k+1} = \sum_{j=0}^{k} W_j u_{k-j} + \sum_{j=0}^{k+1} K_j e_{k+1-j} \qquad (2)$$

with bounded linear operators $W_j : \mathcal{U} \to \mathcal{U}$ and $K_j : \mathcal{Y} \to \mathcal{U}$, regarded as compensation elements and/or filters to condition the signals. Typically $K_j = 0$ (resp. $W_j = 0$) for $j > M_e$ (resp. $j > M_u$). A simple structure is

$$u_{k+1} = W_0 u_k + K_0 e_{k+1} + K_1 e_k \qquad (3)$$

Assuming that G and W_0 commute (i.e., $GW_0 = W_0 G$), the resultant error evolution takes the form

$$e_{k+1} = (I + GK_0)^{-1}(W_0 - GK_1)e_k \\ + (I + GK_0)^{-1}(I - W_0)(r - d)$$

ROBUST ILC: *An ILC algorithm is said to be robust if convergence is retained in the presence of a defined class of modeling errors.*

Results from multipass systems theory (Edwards and Owens 1982) indicate robust convergence of the sequence $\{e_k\}_{k \geq o}$ to a limit $e_\infty \in \mathcal{Y}$ (in the presence of small modeling errors) if the spectral radius condition

$$r[(I + GK_0)^{-1}(W_0 - GK_1)] < 1 \qquad (4)$$

is satisfied where $r[\cdot]$ denotes the spectral radius of its argument. However, the desired condition $e_\infty = 0$ is true only if $W_0 = I$. For a given r, it may be possible to retain the benefits of choosing $W_0 \neq I$ and still ensure that e_∞ is sufficiently small for the application in mind, e.g., by limiting limit errors to a high-frequency band. This and other spectral radius conditions form the underlying convergence condition when choosing controller elements but are rarely computed. The simplest algorithm using eigenvalue

computation for a linear discrete time system defines the *relative degree* to be $k^* = 0$ if $D \neq 0$ and the smallest integer k such that $CA^{k-1}B \neq 0$ otherwise. Replacing $\mathcal{Y}$ by the range of G; choosing $W_0 = I, K_0 = 0$, and $K_1 = I$; and supposing that $k^* \geq 1$, the Arimoto input update rule $u_{k+1}(t) = u_k(t) + e_k(t + k^*)$, $0 \leq t \leq N + 1 - k^*$ provides robust convergence if, and only if, $r[I - CA^{k^*-1}B] < 1$. It does not imply that the error signal necessarily improves each iteration. Errors can reach very high values before finally converging to zero. However, if (4) is replaced by the operator norm condition

$$\|(I + GK_0)^{-1}(W_0 - GK_1)\| < 1, \quad \text{then (5)}$$

$\{\|e_k - e_\infty\|_Q\}_{k \geq 0}$ monotonically decreases to zero.

The spectral radius condition throws light on the nature of ILC robustness. Choosing, for simplicity, $K_0 = 0$ and $W_0 = I$, the requirement that $r[I - GK_1] < 1$ will be satisfied by a wide range of processes G, namely those for which the eigenvalues of $I - GK_1$ lie in the open unit circle of the complex plane. Translating this requirement into useful robustness tests may not be easy in general. The discussion does however show that the behavior of GK_1 must be "sign-definite" to some extent as, if $r[I - GK_1] < 1$, then $r[I - (-G)K_1] > 1$, i.e., replacing the plant by $-G$ (no matter how small) will inevitably produce non-convergent behavior. A more detailed characterization of this property is possible for inverse model ILC.

Inverse Model-Based Iteration

If a linear system G has a well-defined inverse model G^{-1}, then the required input signal is $u_\infty = G^{-1}(r - d)$. The simple update rule

$$u_{k+1} = u_k + \beta G^{-1} e_k, \quad (6)$$

where β is a *learning gain*, produces the dynamics

$$e_{k+1} = (1 - \beta)e_k \quad \text{or} \quad e_{k+1} = (1 - \beta)^k e_0,$$

proving that zero error is attainable with added flexibility in convergence rate control by choosing $\beta \in (0, 2)$. Errors in the system model used in (6) are an issue. Insight into this problem has been obtained for single-input, single-output discrete time systems with multiplicative plant uncertainty U as retention of monotonic convergence is ensured (Owens and Chu 2012) by a frequency domain condition

$$|\frac{1}{\beta} - U(e^{i\theta})| < \frac{1}{\beta}, \text{ for all } \theta \in [0, 2\pi] \quad (7)$$

that illustrates a number of general empirical rules for ILC robust design. The first is that a small learning gain (and hence small input update changes and slow convergence) will tend to increase robustness and, hence, that it is *necessary* that multiplicative uncertainties satisfy some form of strict positive real condition which, for (6), is

$$\text{Re}\left[U(e^{i\theta})\right] > 0, \quad \text{for all } \theta \in [0, 2\pi], \quad (8)$$

a condition that limits high-frequency roll-off error and constrains phase errors to the range $(-\frac{\pi}{2}, \frac{\pi}{2})$. The second observation is that if G is non-minimum phase, the inverse G^{-1} is unstable, a situation that cannot be tolerated in practice.

Optimization-Based Iteration

Design criteria can be strengthened by a monotonicity requirement. Measuring error magnitude by a norm $\|e\|_Q$ on $\mathcal{Y}$, such as the weighted mean square error (with Q symmetric, positive definite)

$$\|e\|_Q = \sqrt{\int_0^T e^T(t)Qe(t)dt},$$

then the condition $\|e_{k+1}\|_Q < \|e_k\|_Q$ for all $k \geq 0$ provides a performance improvement from iteration to iteration. This idea leads to a number of design approaches, Owens and Daley (2008) and Ahn et al. (2007b) (which also examines aspects of robustness).

Function/Time Series Optimization

Norm optimal ILC (NOILC) (Owens and Daley 2008) guarantees monotonicity and convergence to $e_\infty = 0$ by computing u_{k+1} to minimize an objective function

$$J(u) = \|e\|_Q^2 + \|u - u_k\|_R^2,$$

subject to plant dynamics. For linear models (1),

$$u_{k+1} = u_k + G^* e_{k+1}$$

where $G^* : \mathcal{Y} \to \mathcal{U}$ is the adjoint operator of G. For continuous or discrete time linear state space models, the problem is a classical optimal tracking problem with a solution with online state feedback and a feedforward term generated offline by simulation of an "adjoint" model. Reducing R in J leads to faster convergence rates, but the presence of non-minimum-phase zeros has a negative effect on convergence (Owens and Chu 2010). Monotonicity and convergence to zero is retained, but, after an initial fall, the error norm then reduces infinitesimally each iteration producing the practical effect of limited error reductions over finite iteration horizons. Rules exist (Owens and Chu 2010) to minimize the effect by choice of u_0 and r.

Related Linear NOILC Problems

If $\mathcal{Y}$ and $\mathcal{U}$ are real Hilbert spaces, geometrical arguments can be used to generate algorithms extending the NOILC algorithm to include (Owens and Daley 2008) acceleration mechanisms, predictive control, and the inclusion of input signal constraints. They also allow more flexibility in the form and specification of the task. In the intermediate point NOILC problem (denoted IPNOILC), the task requirement is that the output signal $y(t)$, $0 \le t \le T$ takes specified values $r(t_1), r(t_2), \ldots, r(t_M)$ as it passes through the M *intermediate points* $0 < t_1 < t_2 < \cdots < t_M$. The precise nature of the trajectory between points is of secondary importance. Again, the solution, for linear state space systems, can be constructed from Riccati equation-based feedback rules combined with "jump" conditions and feedforward control signals computed off-line.

The IPNOILC solution is nonunique, and the remaining degrees of freedom can be used to satisfy other design objectives. Switching algorithms (Owens et al. 2013) converge to a solution of the problem while simultaneously minimizing an auxiliary criterion

$$J_{\mathrm{aux}}(u) = \|z - z_0\|_{\tilde{Q}}^2 + \|u - u_0\|_R^2.$$

Auxiliary optimization is a tool for shaping the solution of the IPNOILC problem. The auxiliary variable z could be internal states whose behavior is important to plant operation or simply defined by the output, e.g., $z = \ddot{y}$ which, if small, might reduce input "forces" and hence actuator activity.

Parameter Optimization

NOILC can be simplified by reducing the degrees of freedom defining control action to a small number of control law parameters. For a discrete system (1), a general update rule is

$$u_{k+1} = u_k + \Gamma(\beta_{k+1})e_k, \ \ k \ge 0 \ .$$

Here the matrix $\Gamma(\beta)$ is linear in the $p \times 1$ parameter vector β with $\Gamma(0) = 0$. Under these conditions $\Gamma(\beta)e = F(e)\beta$ where the matrix $F(e)$ is linear in e with $F(0) = 0$. Examples of useful parameterizations include inverse model control (Owens et al. 2012).

Monotonicity of the error norm is ensured by choosing the parameter vector β_{k+1} to minimize

$$J(\beta) = \|e\|_Q^2 + \beta^T W_{k+1} \beta$$

subject to the dynamic constraint (1). Each $p \times p$ weighting matrix W_{k+1} is symmetric, positive definite, and may be iteration dependent. The algorithm creates a nonlinear ILC law providing a link between parameter evolution, past errors, and the choice of weight W_{k+1}.

Summary and Future Directions

The basic structure of ILC is now well understood with a number of algorithms available with known convergence properties and empirical

links between parameter choice and convergence rates (Ahn et al. 2007a; Bristow et al. 2006; Owens and Daley 2008; Wang et al. 2009; Xu 2011). Optimization-based algorithms provide a structured approach to convergence and have a familiar quadratic optimal control structure. Despite the practical benefits of monotonic error norms, this approach underlines the difficulties induced by non-minimum-phase (NMP) properties of the plant. Operator representations extend this theory to include more general problems such as the intermediate point tracking problem and, where solutions are non-unique, can be converted into iterative algorithms that inherit the properties of NOILC but converge to a solution that also minimizes an auxiliary optimization criterion.

Many of the challenges addressed by NOILC are inherited by other algorithms, many of which mimic established control design paradigms. For example, the commonly used PD update law

$$u_{k+1}(t) = u_k(t) + K_1 e_k(t) + K_2 \dot{e}_k(t)$$

can produce convergence by suitable choice of K_1 and K_2. Proofs of convergence are typically based on spectral radius conditions similar to (4) for linear systems or on techniques such as contraction mapping (fixed point) theorems (Xu 2011) for nonlinear systems. The nonlinear case generally suggests *local* convergence conditions dependent on growth conditions on the nonlinearity. They typically cannot be checked in practice but do link convergence to simple, empirical, gain selection rules.

ILC, as a topic, is a very large area of study. Survey papers indicate that progress has been made in a number of other areas including adaptive ILC, the use of intelligent control ideas of fuzzy logic and neural networks-based control structures, $2D$ systems theory, and mathematical studies of fractional order control laws (Chen et al. 2013). The further development of ILC from its current strong base will draw extensively from classical control knowledge but relies on the three aspects of *plant modeling*, *control design*, and *coping with uncertainty*. Issues central to medium-term success include:

1. Extending current ILC knowledge to other classes of model needed for applications.
2. Integration of online data-based modeling into ILC schemes to enhance adaptive control options.
3. Ensuring the property of error monotonicity or characterizing any non-monotonicity to be expected.
4. The construction of robustness tests and using the ideas in new robust design methodologies.
5. Providing a better understanding of the effect of noise and disturbances on algorithm performance.
6. Extending the range of tasks to include, for example, different challenges for different outputs on different subintervals of $[0, T]$.
7. Creating design tools for nonlinear plant that ensure convergence and a degree of robustness but, in particular, provide some control of internal plant states that may be subject to dynamical constraints.

Cross-References

▶ Adaptive Control, Overview
▶ Adaptive Control of Linear Time-Invariant Systems
▶ Generalized Finite-Horizon Linear-Quadratic Optimal Control
▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions
▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions
▶ Linear Quadratic Optimal Control
▶ Optimal Control and Pontryagin's Maximum Principle

Bibliography

Ahn H-S, Chen YQ, Moore KL (2007a) Iterative learning control: brief survey and categorization. IEEE Trans Syst Man Cybern C Appl Rev 37(3):1099–1121
Ahn H-S, Moore KL, YQ Chen (2007b) Iterative learning control: robustness and monotonic convergence for interval systems. Series on communications and control engineering. Springer, New York/London

Arimoto S, Miyazaki F, Kawamura S (1984) Bettering operation of robots by learning. J Robot Syst 1(2): 123–140

Bien Z, Xu J-X (1998) Iterative learning control: analysis, design, integration and applications. Kluwer Academic, Boston

Bristow DA, Tharayil M, Alleyne AG (2006) A survey of iterative learning control a learning-based method for high-performance tracking control. IEEE Control Syst Mag 26(3):96–114

Chen Y, Wen C (1999) Iterative learning control: convergence, robustness and applications. Lecture notes in control and information sciences, vol 48. Springer, London/New York

Chen YQ, Li Y, Ahn H-S, Tian G (2013) A survey on fractional order iterative learning. J Optim Theory Appl 156:127–140

Daley S, Owens DH, Hatonen J (2007) Application of optimal iterative learning control to the dynamic testing of mechanical structures. Proc Inst Mech Eng I J Syst Control Eng 221:211–222

Edwards JB, Owens DH (1982) Analysis and control of multipass processes. Research Studies Press/Wiley, Chichester

Owens DH, Chu B (2010) Modelling of non-minimum phase effects in discrete-time norm optimal iterative learning control. Int J Control 83(10): 2012–2027

Owens DH, Chu B (2012) Combined inverse and gradient iterative learning control: performance, monotonicity, robustness and non-minimum-phase zeros. Int J Robust Nonlinear Control (2):203–226. doi:10:1002/rnc.2893

Owens DH, Daley S (2008) Iterative learning control – monotonicity and optimization. Int J Appl Math Comput Sci 18(3):279–293

Owens DH, Chu B, Songjun M (2012) Parameter-optimal iterative learning control using polynomial representations of the inverse plant. Int J Control 85(5): 533–544

Owens DH, Freeman CT, Chu B (2013) Multivariable norm optimal iterative learning control with auxiliary optimization. Int J Control 1(1):1–2

Rogers E, Galkowski K, Owens DH (2007) Control systems theory and applications for linear repetitive processes. Lecture notes in control and information sciences, vol 349. Springer, Berlin

Rogers E, Owens DH, Werner H, Freeman CT, Lewin PL, Schmidt C, Kirchhoff S, Lichtenberg G (2010) Norm-optimal iterative learning control with application to problems in acceleration-based free electron lasers and rehabilitation robotics. Eur J Control 5:496–521

Wang Y, Gao F, Doyle FJ III (2009) Survey on iterative learning control repetitive control and run-to run control. J Process Control 19:1589–1600

Xu J-X (2011) A survey on iterative learning control for nonlinear systems. Int J Control 84(7):1275–1294

K

Kalman Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Abstract

The Kalman filter is a very useful algorithm for linear Gaussian estimation problems. It is extremely popular and robust in practical applications. The algorithm is easy to code and test. There are many reasons for the popularity of the Kalman filter in the real world, including stability and generality and simplicity. Moreover, the real-time computational complexity is very reasonable for high-dimensional problems. In particular, the computational complexity scales as the cube of the dimension of the state vector.

Keywords

Controllability; Discrete time measurements; Estimation algorithm; Extended Kalman filter; Filtering; Gaussian errors; Linear dynamical system; Linear system; Observability; Recursive; Smoothing; Stability

Description of Kalman Filter

The Kalman filter is an algorithm that computes the best estimate of the state vector of a linear dynamical system given discrete time measurements of a linear function of the state vector corrupted by additive white Gaussian noise. The Kalman filter also quantifies the uncertainty in its estimate of the state vector, using the covariance matrix of estimation errors. The detailed equations of the Kalman filter algorithm and the problem that it solves are given in Gelb et al. (1974), which is the most accessible but thorough book on Kalman filters. The linear dynamical system can be time varying, but its parameters must be known exactly. The measurements can be made at arbitrary (nonuniform) discrete times, but these times must be known exactly. Likewise, the covariance matrices of the measurement errors and the process noise can be arbitrary and time varying, but the numerical values of these covariance matrices must be known exactly. Also, the initial uncertainty in the state vector must be Gaussian and the mean and covariance matrix can be arbitrary, but these must be known exactly. There is a very powerful theory of Kalman filter stability due to Kalman (1963), which guarantees that the Kalman filter is stable under very mild technical assumptions which can always be satisfied in practice. In particular, the Kalman filter is stable for estimating the state vector of linear dynamical systems that are stable or unstable, for arbitrarily slow measurement rates, provided that the mild technical assumptions are fulfilled. These assumptions require that the dimension of the state vector is minimal and that the measurement error covariance matrix and the process noise covariance matrices are positive

definite, although weaker conditions are also sufficient for stability in some cases; see Kailath et al. (2000) for such details on the stability of the Kalman filter. Kalman filter stability is connected with observability and controllability of the input-output model of the relevant dynamical system in Kalman (1963) The corresponding algorithm for continuous time linear measurements (with Gaussian additive white noise) and continuous time linear dynamical systems (with Gaussian additive white process noise) is called the Kalman-Bucy filter; see Kalman (1961).

Design Issues

In engineering practice, almost all real-world applications are nonlinear or non-Gaussian, and therefore, they do not fit the Kalman filter theory. Nevertheless, by approximating the nonlinear dynamics and measurements with linear equations, one can apply the Kalman filter theory; this is called the "extended Kalman filter" (EKF); see Gelb et al. (1974). The linearization of the nonlinear dynamics and measurements is made by computing the first-order Taylor series expansion and evaluating it at the estimated state vector; this is a very simple and fast approximation that is widely used in real- world applications, and it often gives good estimation accuracy, although there is no guarantee of that. Moreover, there is no guarantee that the EKF will be stable, even if the linearized system satisfies all the theoretical requirements for stability of the Kalman filter.

Even if the dynamics and measurements are exactly linear and if the measurement noise and process noise and initial uncertainty are all exactly Gaussian with exactly known means and covariance matrices, there can still be significant practical problems with Kalman filter accuracy, owing to ill-conditioning. In particular, the Kalman filter can be extremely sensitive to quantization errors in the computer arithmetic and storage. On the other hand, there are many different methods to try to mitigate ill-conditioning including (1) double or quadruple or octuple precision arithmetic, (2) making the covariance matrices symmetric before and after

every operation, (3) Tychonov regularization, (4) tuning the process noise covariance matrix, (5) coding the Kalman filter in principal coordinates or approximately principal coordinates (i.e., aligned with the eigenvectors of the state vector error covariance matrix), (6) sequential scalar measurement updates in a preferred order, and (7) various factorizations of the covariance matrices (e.g., square root, information matrix, information square root, upper triangular and lower triangular factorization, UDL, etc.). The classic book on error covariance matrix factorizations is by Bierman (2006). Unfortunately, there is no guarantee that the Kalman filter will work well even if all of these mitigation methods are used. Moreover, there is no useful theoretical analysis of this phenomenon, with the exception of a few not very tight upper bounds on the condition number. Plotting the numerical values of the condition number of the covariance matrix vs. time is often a helpful diagnostic. In certain real-world applications, the condition number of the Kalman filter error covariance matrix can be ten billion or larger.

Why Is the Kalman Filter So Useful and Popular?

The Kalman filter has been enormously successful in real-world applications, and it is interesting to reflect on why it has been so useful and so popular. In particular, Kalman himself believes that his filter was successful because it was based on probability rather than statistics; see Kalman (1978). For example, the error covariance matrix for the Kalman filter is computed from the assumed dynamics and measurement model and the assumed values of the initial state uncertainty and process noise and measurement noise covariance matrices, rather than by computing sample covariance matrices. Likewise, the Kalman filter computes the estimated state vector from assumed Gaussian and linear probability models, rather than computing the sample mean, as would be done in statistics. There is substantial wisdom in Kalman's assertion, owing to the difficulty of estimating sample covariance matrices

and sample vectors that are sufficiently accurate, given a limited number of samples and ill-conditioning and high-dimensional state vectors. A second reason that Kalman filters are so popular is that the real-time computational complexity is very reasonable for modern digital computers, even for problems with a high-dimensional state vector. In particular, the computational complexity of the Kalman filter scales as the cube of the dimension of the state vector; for example, the modern GPS system uses a Kalman filter with a state vector of dimension of about 1,000 to jointly estimate the orbits of the satellites in the GPS constellation. In 1960, when Kalman's paper was first published, digital computers were starting to become fast enough at reasonable cost to multiply large matrices in real time, which is the most challenging computation in a Kalman filter. Today computers are roughly ten orders of magnitude faster per unit cost than in 1960, and hence, we can run Kalman filters for high-dimensional problems on very inexpensive computers that fit into your wristwatch. A third reason that Kalman filters are popular is that the algorithms are easy to understand and code and test. A fourth reason is the guaranteed stability of the Kalman filter under very mild conditions which can always be satisfied in practical applications. A fifth reason is that the Kalman filter is optimal for time-varying unstable linear dynamics with time-varying measurement noise covariance and process noise covariance. A sixth reason is that one can use Kalman filters for nonlinear problems by approximating the nonlinear dynamics and measurement equations with a first-order Taylor series; this is called the extended Kalman filter (EKF), which is without a doubt the most widely used algorithm in real-world estimation applications. A seventh reason is that the Kalman filter automatically provides a convenient quantification of uncertainty of the estimated state vector using the error covariance matrix. The final reason is that it is easy to test the accuracy of the Kalman filter by comparing the theoretical error covariance matrix to errors computed by Monte Carlo simulations of the filter; the two errors should agree approximately, and statistically significant discrepancies suggest

bugs in the code or ill-conditioning of the error covariance matrices or nonlinearities or errors in modeling the dynamics or measurements.

Kalman's 1960 paper represented a big paradigm shift in two ways: (1) it exploited fast low-cost modern digital computers, whereas the literature up to that time did not, and (2) it used time domain methods rather than the ubiquitous Fourier transform methods, which limited the dynamics to steady state asymptotic in time, which in turn limited the theory to cover stable dynamics. Of course today we take both of these big points as normal engineering rather than revolutionary and surprising. The state of the art prior to Kalman's 1960 paper was the Wiener filter, which was based firmly on the Fourier transform. The Wiener filter required very lengthy and cumbersome algebraic spectral factorization with complex variables, resulting in erroneous formulas published in certain books, owing to algebraic errors which were not obvious and which could not be checked by computers, owing to the nonexistence of computer algebra software (e.g., MATHEMATICA) in 1960. Kalman explains many other problems with the Wiener filter in Kalman (2003).

Summary and Future Directions

To a large extent the Kalman filter theory is complete. There is a simple and useful theory of stability of the Kalman filter, which is completely lacking for nonlinear filters including the extended Kalman filter (EKF) and particle filters. Moreover, there are many robust versions of the Kalman filter that have been invented to mitigate ill-conditioning of the covariance matrix as well as uncertainty in system models and system parameters. However, there remain many important design issues that need to be addressed for practical applications; see Daum (2005) for details. The most obvious issues include nonlinear measurements, nonlinear plant models, robustness to uncertainty in measurement models and plant models, non-Gaussian measurement noise and plant noise, non-Gaussian initial uncertainty in the state vector, and ill-conditioning

of the error covariance matrix. That is, any deviation from the exact mathematical assumptions in the Kalman filter theory can cause problems in practice. It is the job of engineers to mitigate such problems and design filters that are robust to such perturbations. Kalman's recent papers on how the Kalman filter was invented and why it is so popular contain interesting and useful ideas; see Kalman (1978) and Kalman (2003).

Cross-References

▶ Estimation, Survey on
▶ Extended Kalman Filters
▶ Nonlinear Filters

Bibliography

Bierman G (2006) Factorization methods for discrete sequential estimation. Dover, Mineola
Daum F (2005) Nonlinear filters: beyond the Kalman filter. IEEE AES Mag 20:57–69
Gelb A et al (1974) Applied optimal estimation. MIT, Cambridge
Kailath T (1974) A view of three decades of linear filtering theory. IEEE Trans Inf Theory 20:146–181
Kailath T, Sayed A, Hassibi B (2000) Linear estimation. Prentice Hall, Upper Saddle River
Kalman R (1960) A new approach to linear filtering and prediction problems. Trans ASME J 82:35–45
Kalman R (1961) New methods in Wiener filtering theory. RIAS technical report 61-1 (Feb 1961); also published in Bogdanoff J, Kozin F (eds) (1963) Proceedings of first symposium on engineering applications of random function theory and probability. Wiley, pp 270–388
Kalman R (1978) A retrospective after twenty years: from the pure to the applied. In: Applications of the Kalman filter to hydrology, edited by Chao-lin Chiu, University of Pittsburgh, LC card number 78-069752, available on-line. pp 31–54
Kalman R (2003) Discovery and invention: the Newtonian revolution in systems technology. AIAA J Guid Control Dyn 26(6):833–837
Kalman R, Bucy R (1961) New results in linear filtering and prediction theory. Trans ASME 83:95–107
Sorenson H (1970) Least squares estimation from Gauss to Kalman. IEEE Spectr 7:63–68
Stepanov OA (2011) Kalman filtering: past and present. J Gyroscopy Navig 2:99–110

KYP Lemma and Generalizations/Applications

Tetsuya Iwasaki
Department of Mechanical & Aerospace Engineering, University of California, Los Angeles, CA, USA

Abstract

Various properties of dynamical systems can be characterized in terms of inequality conditions on their frequency responses. The Kalman-Yakubovich-Popov (KYP) lemma shows equivalence of such frequency domain inequality (FDI) and a linear matrix inequality (LMI). The fundamental result has been a basis for robust and optimal control theories in the past several decades. The KYP lemma has recently been generalized to the case where an FDI on a possibly improper transfer function is required to hold in a (semi)finite frequency range. The generalized KYP lemma allows us to directly deal with practical situations where design parameters are sought to satisfy FDIs in multiple (semi)finite frequency ranges. Various design problems, including FIR filter and PID controller, reduce to LMI problems which can be solved via semidefinite programming.

Keywords

Bounded real; Frequency domain inequality; Linear matrix inequality; Multi-objective design; Optimal control; Positive real; Robust control

Introduction

In linear systems analysis and control design, dynamical properties are often characterized by frequency responses. The shape of a frequency response, as visualized by the Bode or Nyquist plot, is closely related to various performance measures including the

steady state error, fast and smooth transient, and robustness against unmodeled dynamics. Hence, desired system properties can be formalized in terms of a set of frequency domain inequalities (FDIs) on selected transfer functions. The analysis and design problems then reduce to verification and satisfaction of the FDIs.

The Kalman-Yakubovich-Popov (KYP) lemma (Anderson 1967; Kalman 1963; Rantzer 1996; Willems 1971) establishes the equivalence between an FDI and a linear matrix inequality (LMI). The LMI is defined by state space matrices of the transfer function in the FDI so that the FDI holds true if and only if the LMI admits a solution. The LMI characterization of an FDI is useful since it replaces the process of checking the FDI at infinitely many frequency points by the search for a symmetric matrix satisfying a finite dimensional convex constraint defined by the LMI. In addition to exact and tractable computations, benefits of the LMI conditions include analytical understanding of robust and optimal controls through spectral factorizations and storage/Lyapunov functions. The KYP lemma is a fundamental result in the systems and control field that has provided, in the past half century, a theoretical basis for developments of various tools for system analysis and design.

A drawback of the KYP lemma is its inability to characterize an FDI in a finite frequency range. Feedback control designs typically involve a set of specifications given in terms of multiple FDIs in various frequency ranges. However, the KYP lemma is not capable of treating such FDIs directly since it has to consider the entire frequency range. To address this deficiency, the KYP lemma has recently been generalized to characterize an FDI in a finite frequency range exactly (Iwasaki et al. 2000). Further generalizations (Iwasaki and Hara 2005) are available for FDIs within various frequency ranges for both continuous- and discrete-time, possibly improper, rational transfer functions. The generalized KYP lemma allows for direct multiobjective design of filters, controllers, and dynamical systems.

KYP Lemma

The KYP lemma may be motivated from various aspects, but let us explain it as an extension of a gain condition. Consider a stable linear system

$$\dot{x} = Ax + Bu, \quad G(s) := (sI - A)^{-1}B,$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, and $G(s)$ is the transfer function from u to x. If u is a disturbance to the system and x represents the error from a desired operating point, we may be interested in how large the state variables can become for a given magnitude of the disturbance. The gain $\|G(j\omega)\|$ captures this property for the case of a sinusoidal disturbance at frequency ω, where $\|\cdot\|$ denotes the spectral norm ($=$ absolute value for a scalar). If $\|G(j\omega)\| < \gamma$ holds for all frequency ω with a small γ, then the system has a good disturbance attenuation property.

A version of the KYP lemma states that the FDI $\|G(j\omega)\| < \gamma$ with $\gamma = 1$ holds for all frequency ω if and only if there exists a symmetric matrix P satisfying the LMI:

$$\begin{bmatrix} PA + A^\mathsf{T}P + I & PB \\ B^\mathsf{T}P & -I \end{bmatrix} < 0.$$

Thus, existence of one particular P satisfying the LMI is enough to conclude that the gain is less than one for all, infinitely many, frequencies. This result is known as the bounded real lemma and has played a fundamental role in the robust and H_∞ control theories.

The KYP lemma can be introduced as a generalization of the bounded real lemma. First, note that the gain bound condition $\|G(j\omega)\| < 1$ and the LMI condition can equivalently be written as

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Theta \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} < 0, \qquad (1)$$

$$\begin{bmatrix} PA + A^\mathsf{T}P & PB \\ B^\mathsf{T}P & 0 \end{bmatrix} + \Theta < 0, \qquad (2)$$

where

$$\Theta := \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}.$$

In these equations, the particular matrix Θ is chosen to describe the gain bound condition as a special case of the quadratic form (1), and we observe that Θ appears in the LMI as in (2). It turns out that the equivalence of (1) and (2) holds not only for this particular Θ but also for an arbitrary symmetric matrix Θ. This result is called the KYP lemma, which states that, given arbitrary matrices A, B, and $\Theta = \Theta^\top$, the FDI (1) holds for all frequency ω if and only if there exists a matrix $P = P^\top$ satisfying the LMI (2), provided A has no eigenvalues on the imaginary axis.

The FDI in (1) can be specialized to an FDI

$$\begin{bmatrix} L(j\omega) \\ I \end{bmatrix}^* \Pi \begin{bmatrix} L(j\omega) \\ I \end{bmatrix} < 0. \qquad (3)$$

on transfer function

$$L(s) := C(sI - A)^{-1}B + D$$

by choosing

$$\Theta := \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^\top \Pi \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}. \qquad (4)$$

The choice of matrix Π allows for characterizations of important system properties involving gain and phase of $L(s)$. For instance, the FDI (3) with

$$\Pi := \begin{bmatrix} 0 & -I \\ -I & 0 \end{bmatrix}$$

gives $L(j\omega) + L(j\omega)^* > 0$. This is called the positive real property, with which the phase angle remains between $\pm 90°$ when $L(j\omega)$ is a scalar.

Generalization

The standard KYP lemma deals with FDIs that are required to hold for all frequencies. To allow for more flexibility in practical system designs, the KYP lemma has been generalized to deal with FDIs in (semi)finite frequency ranges.

For instance, a version of the generalized KYP lemma states that the FDI (1) holds in the low frequency range $|\omega| \leq \varpi_\ell$ if and only if there exist matrices $P = P^\top$ and $Q = Q^\top > 0$ satisfying

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^\top \begin{bmatrix} -Q & P \\ P & \varpi_\ell^2 Q \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \Theta < 0, \qquad (5)$$

provided A has no imaginary eigenvalues in the frequency range. In the limiting case where ϖ_ℓ approaches infinity and the FDI is required to hold for the entire frequency range, the solution Q to (5) approaches zero, and we recover (2).

The role of the additional parameter Q is to enforce the FDI only in the low frequency range. To see this, consider the case where the system is stable and a sinusoidal input $u = \Re[\hat{u}e^{j\omega t}]$, with (complex) phasor vector $\hat{u}$, is applied. The state converges to the sinusoid $x = \Re[\hat{x}e^{j\omega t}]$ in the steady state where $\hat{x} := G(j\omega)\hat{u}$. Multiplying (5) by the column vector obtained by stacking $\hat{x}$ and $\hat{u}$ in a column from the right, and by its complex conjugate transpose from the left, we obtain

$$(\varpi_\ell^2 - \omega^2)\hat{x}^* Q \hat{x} + \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix}^* \Theta \begin{bmatrix} \hat{x} \\ \hat{u} \end{bmatrix} < 0.$$

In the low frequency range $|\omega| \leq \varpi_\ell$, the first term is nonnegative, enforcing the second term to be negative, which is exactly the FDI in (1). If ω is outside of the range, however, the first term is negative, and the FDI is not required to hold.

Similar results hold for various frequency ranges. The term involving Q in (5) can be expressed as the Kronecker product $\Psi \otimes Q$ with Ψ being a diagonal matrix with entries $(-1, \varpi_\ell^2)$. The matrix Ψ arises from characterization of the low frequency range:

$$\begin{bmatrix} j\omega \\ 1 \end{bmatrix}^* \Psi \begin{bmatrix} j\omega \\ 1 \end{bmatrix} = \varpi_\ell^2 - \omega^2 \geq 0.$$

By different choices of Ψ, middle and high frequency ranges can also be characterized:

	Low	Middle	High
$\boldsymbol{\Omega}$	$\lvert\omega\rvert \leq \varpi_\ell$	$\varpi_1 \leq \omega \leq \varpi_2$	$\lvert\omega\rvert \geq \varpi_h$
Ψ	$\begin{bmatrix} -1 & 0 \\ 0 & \varpi_\ell^2 \end{bmatrix}$	$\begin{bmatrix} -1 & j\varpi_c \\ -j\varpi_c & -\varpi_1\varpi_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -\varpi_h^2 \end{bmatrix}$

where $\varpi_c := (\varpi_1 + \varpi_2)/2$ and $\boldsymbol{\Omega}$ is the frequency range. For each pair $(\boldsymbol{\Omega}, \Psi)$, the FDI (1) holds in the frequency range $\omega \in \boldsymbol{\Omega}$ if and only if there exist real symmetric matrices P and $Q > 0$ satisfying

$$F^{\mathsf{T}}(\Phi \otimes P + \Psi \otimes Q)F + \Theta < 0, \quad (6)$$

provided A has no eigenvalues in $\boldsymbol{\Omega}$, where

$$\Phi := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad F := \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}.$$

Further generalizations are available Iwasaki and Hara (2005). The discrete-time case (frequency variable on the unit circle) can be similarly treated by a different choice of Φ. FDIs for descriptor systems and polynomial (rather than rational) functions can also be characterized in a form similar to (6) by modifying the matrix F. More specifically, the choices

$$\Phi := \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad F := \begin{bmatrix} A & B \\ E & O \end{bmatrix}$$

give the result for the discrete-time transfer function $L(z) = (zE - A)^{-1}(B - zO)$.

Applications

The generalized KYP lemma is useful for a variety of dynamical system designs. As an example, let us consider a classical feedback control design via shaping of a scalar open-loop transfer function in the frequency domain. The objective is to design a controller $K(s)$ for a given plant $P(s)$ such that the closed-loop system is stable and possesses a good performance dictated by reference tracking, disturbance attenuation, noise sensitivity, and robustness against uncertainties.
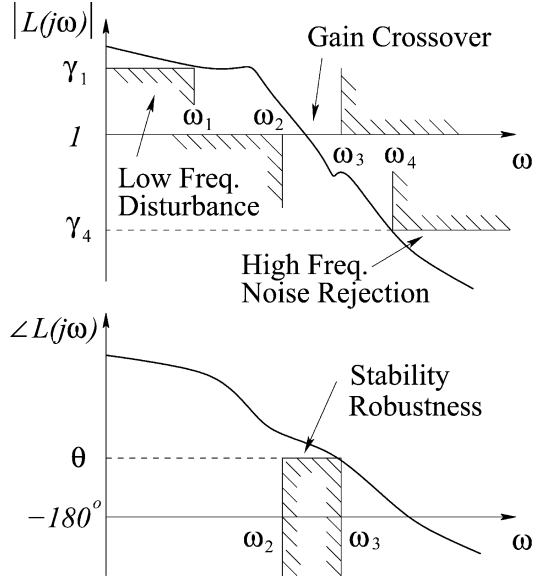
KYP Lemma and Generalizations/Applications, Fig. 1 Loop shaping design specifications

Typical design specifications are given in terms of bounds on the gain and phase of the open-loop transfer function $L(s) := P(s)K(s)$ in various frequency ranges as shown in Fig. 1. The controller $K(s)$ should be designed so that the frequency response $L(j\omega)$ avoids the shaded regions. For instance, the gain should satisfy $\lvert L(j\omega)\rvert \geq 1$ for $\lvert\omega\rvert < \omega_2$ and $\lvert L(j\omega)\rvert \leq 1$ for $\lvert\omega\rvert > \omega_3$ to ensure the gain crossover occurs in the range $\omega_2 \leq \omega \leq \omega_3$, and the phase bound $\angle L(j\omega) \geq \theta$ in this range ensures robust stability by the phase margin.

The design specifications can be expressed as FDIs of the form (3), where a particular gain or phase condition can be specified by setting Π as

$$\pm \begin{bmatrix} 1 & 0 \\ 0 & -\gamma^2 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & j - \tan\theta \\ -j - \tan\theta & 0 \end{bmatrix}$$

with $\gamma = \gamma_1$, γ_4, or 1, and the $+/-$ signs for upper/lower gain bounds. These FDIs in the corresponding frequency ranges can be converted to inequalities of the form (6) with Θ given by (4).

The control problem is now reduced to the search for design parameters satisfying the set of

inequality conditions (6). In general, both coefficient matrices F and Θ may depend on the design parameters, but if the poles of the controller are fixed (as in the PID control), then the design parameters will appear only in Θ. If in addition an FDI specifies a convex region for $L(j\omega)$ on the complex plane, then the corresponding inequality (6) gives a convex constraint on P, Q, and the design parameters. This is the case for specifications of gain upper bound (disk: $|L| < \gamma$) and phase bound (half plane: $\theta \leq \angle L \leq \theta + \pi$). A gain lower bound $|L| > \gamma$ is not convex but can often be approximated by a half plane. The design parameters satisfying the specifications can then be computed via convex programming.

Various design problems other than the open-loop shaping can also be solved in a similar manner, including finite impulse response (FIR) digital filter design with gain and phase constraints in a passband and stop-band and sensor or actuator placement for mechanical control systems (Hara et al. 2006; Iwasaki et al. 2003). Control design with the Youla parametrization also falls within the framework if a basis expansion is used for the Youla parameter and the coefficients are sought to satisfy convex constraints on closed-loop transfer functions.

Summary and Further Directions

The KYP lemma has played a fundamental role in systems and control theories, equivalently converting an FDI to an LMI. Dynamical systems properties characterized in the frequency domain are expressed in terms of state space matrices without involving the frequency variable. The resulting LMI condition has been found useful for developing robust and optimal control theories.

A recent generalization of the KYP lemma characterizes an FDI for a possibly improper rational function in a (semi)finite frequency range. The result allows for direct solutions of practical design problems to satisfy multiple specifications in various frequency ranges. A design problem is essentially solvable when transfer functions are affine in the design parameters and are required

to satisfy convex FDI constraints. An important problem, which falls outside of this framework and remains open, is the design of feedback controllers to satisfy multiple FDIs on closed-loop transfer functions in various frequency ranges. There have been some attempts to address this problem, but none of them has so far succeeded to give an exact solution.

The KYP lemma has been extended in other directions as well, including FDIs with frequency-dependent weights (Graham and de Oliveira 2010), internally positive systems (Tanaka and Langbort 2011), full rank polynomials (Ebihara et al. 2008), real multipliers (Pipeleers and Vandenberghe 2011), a more general class of FDIs (Gusev 2009), multidimensional systems (Bachelier et al. 2008), negative imaginary systems (Xiong et al. 2012), symmetric formulations for robust stability analysis (Tanaka and Langbort 2013), and multiple frequency intervals (Pipeleers et al. 2013). Extensions of the KYP lemma and related S-procedures are thoroughly reviewed in Gusev and Likhtarnikov (2006). A comprehensive tutorial of robust LMI relaxations is provided in Scherer (2006) where variations of the KYP lemma, including the generalized KYP lemma as a special case, are discussed in detail.

Cross-References

▶ Classical Frequency-Domain Design Methods
▶ H-Infinity Control
▶ LMI Approach to Robust Control

Bibliography

Anderson B (1967) A system theory criterion for positive real matrices. SIAM J Control 5(2): 171–182

Bachelier O, Paszke W, Mehdi D (2008) On the Kalman-Yakubovich-Popov lemma and the multidimensional models. Multidimens Syst Signal Proc 19(3–4):425–447

Ebihara Y, Maeda K, Hagiwara T (2008) Generalized S-procedure for inequality conditions on one-vector-lossless sets and linear system analysis. SIAM J Control Optim 47(3):1547–1555

Graham M, de Oliveira M (2010) Linear matrix inequality tests for frequency domain inequalities with affine multipliers. Automatica 46:897–901

Gusev S (2009) Kalman-Yakubovich-Popov lemma for matrix frequency domain inequality. Syst Control Lett 58(7):469–473

Gusev S, Likhtarnikov A (2006) Kalman-Yakubovich-Popov lemma and the S-procedure: a historical essay. Autom Remote Control 67(11):1768–1810

Hara S, Iwasaki T, Shiokata D (2006) Robust PID control using generalized KYP synthesis. IEEE Control Syst Mag 26(1):80–91

Iwasaki T, Hara S (2005) Generalized KYP lemma: unified frequency domain inequalities with design applications. IEEE Trans Autom Control 50(1):41–59

Iwasaki T, Meinsma G, Fu M (2000) Generalized S-procedure and finite frequency KYP lemma. Math Probl Eng 6:305–320

Iwasaki T, Hara S, Yamauchi H (2003) Dynamical system design from a control perspective: finite frequency positive-realness approach. IEEE Trans Autom Control 48(8):1337–1354

Kalman R (1963) Lyapunov functions for the problem of Lur'e in automatic control. Proc Natl Acad Sci 49(2):201–205

Pipeleers G, Vandenberghe L (2011) Generalized KYP lemma with real data. IEEE Trans Autom Control 56(12):2940–2944

Pipeleers G, Iwasaki T, Hara S (2013) Generalizing the KYP lemma to the union of intervals. In: Proceedings of European control conference, Zurich, pp 3913–3918

Rantzer A (1996) On the Kalman-Yakubovich-Popov lemma. Syst Control Lett 28(1):7–10

Scherer C (2006) LMI relaxations in robust control. Eur J Control 12(1):3–29

Tanaka T, Langbort C (2011) The bounded real lemma for internally positive systems and H-infinity structured static state feedback. IEEE Trans Autom Control 56(9):2218–2223

Tanaka T, Langbort C (2013) Symmetric formulation of the S-procedure, Kalman-Yakubovich-Popov lemma and their exact losslessness conditions. IEEE Trans Autom Control 58(6): 1486–1496

Willems J (1971) Least squares stationary optimal control and the algebraic Riccati equation. IEEE Trans Autom Control 16:621–634

Xiong J, Petersen I, Lanzon A (2012) Finite frequency negative imaginary systems. IEEE Trans Autom Control 57(11):2917–2922

K

L

Lane Keeping

Paolo Falcone
Department of Signals and Systems,
Mechatronics Group, Chalmers University of
Technology, Göteborg, Sweden

Abstract

This chapter provides an overview of lane keeping systems. First, a general architecture is introduced and existing solutions for the necessary sensors and actuators are then overviewed. The threat assessment and the lane position control problems are discussed, highlighting challenges and solutions implemented in lane keeping systems available on the market.

Keywords

Active safety; Decision-making and control; Intelligent transportation systems; Threat assessment

Introduction

Lane keeping systems are vehicle guidance systems that aim at preventing lane departure maneuvers, which may lead to accidents, i.e., collision with surrounding obstacles and vehicles.

By resorting to radar and/or lasers and cameras, a lane keeping system monitors the adjacent lanes. Crossing the lane markings in the absence of vehicles and/or obstacles in the adjacent lanes should not cause any reaction of the lane keeping systems and let the driver freely perform the lane change maneuver. In the presence of vehicles or obstacles in the adjacent lanes, the system should assess the threat and, in case a risk of collision is detected, either warn the driver or automatically issue either a steering or a single-wheel braking command, in order to prevent the crossing of the lane markings. As discussed next, despite the simplicity of the threat assessment and the decision-making and control problem, challenges arise in real traffic scenarios which may lead to nuisance due to unnecessary warnings and/or assisting interventions.

In this entry, we overview the most important aspects in the design of a lane keeping system. This entry is structured as follows. Section "Lane Keeping Systems Architecture" illustrates a generic architecture. Section "Sensing and Actuation" reviews the most used sensors suitable for lane keeping applications. Section "Decision Making and Control" introduces the threat assessment and the lane position control problems, highlighting the most relevant challenges.

Lane Keeping Systems Architecture

The main components of a lane keeping system and their interconnections are shown in Fig. 1.
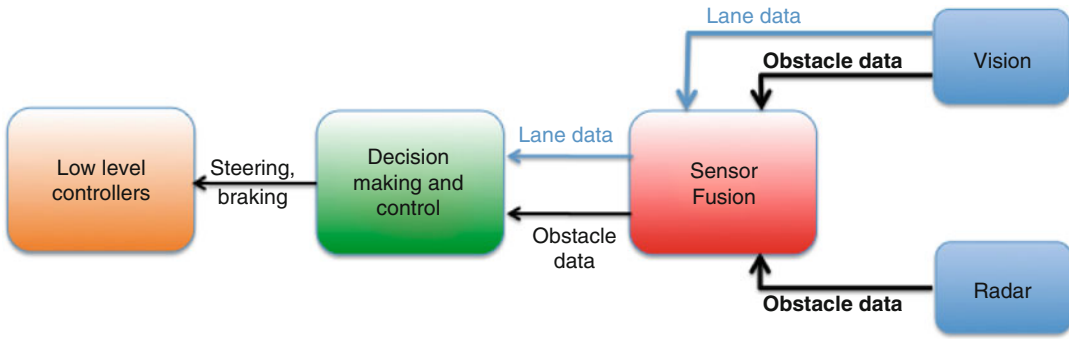
Lane Keeping, Fig. 1 The lane keeping architecture

Relative positions and velocities of the host vehicle w.r.t. the surrounding environment are measured by one radar, typically installed on the front of the vehicle, and possibly by the camera, typically installed on the windshield. Position of the host vehicle within the lane and further information, e.g., road geometry, are measured by the camera. These measurements are then fused by the *sensor fusion* module to provide accurate measurements of the position and velocity of the vehicle w.r.t. the surrounding environment and the lane in the widest range of operating conditions and scenarios.

The task of the *decision-making and control module* is to assess the risk that the vehicle crosses the lane in a dangerous way and, possibly, to take an action that can range from warning the driver or issuing an assisting intervention, e.g., braking and/or steering. Such steering and braking commands are actually implemented by *low-level controllers*.

The different modules will be overviewed in the following sections.

Sensing and Actuation

Radar
Radars for automotive applications are placed in the front of the car, typically behind the grille. The radar emits radio waves and distance from the vehicle ahead is calculated by measuring the arrival time and direction of the reflected radio waves. The relative velocity is determined by relying on the Doppler effect, i.e., by measuring the frequency change of the reflected waves. Relative distance and velocity measurements are typically updated with a frequency of 10 Hz.

Radars for automotive applications emit waves with a frequency of 77 GHz and detect objects within an approximate range of 150 m and a view angle of about $\pm 10°$, with a deviation of 20–30 cm from the correct value for 95 % of the measurements (Eidehall 2004). New radar systems increase the range up to about 200 m with a view angle of about $\pm 10°$ (News Releases DENSO Corporation 2013a).

Typically, radar units are equipped with computer systems running signal processing algorithms that detect and track objects and, for each of them, calculate relative position and speed, azimuth angle, also providing additional information, e.g., the time an object has been tracked and a flag indicating that a target has been locked. Such additional information are typically used in logics implementing the decision-making algorithms of, among others, the lane keeping system.

There are several issues arising from the use of a radar in automotive applications, e.g., wave reflections due to road bumps and barriers that may induce the signal processing algorithms to false object detections (Eidehall 2004). Moreover, interference and the vehicle dynamics (News Releases DENSO Corporation 2013a), e.g., pitching due to braking, may limit the capability of the signal processing algorithms of correctly detecting and tracking the surrounding objects. The latter may be solved by, e.g., using electric motors that

adjust the radar antenna axes in order to compensate for the vehicle dynamics (News Releases DENSO Corporation 2013a).

Vision Systems

Vision systems in lane keeping applications are typically based on a single, CCD camera mounted next to the rear-view mirror placed at the center of the windshield. The image is typically captured by 640×480 pixels and then processed by an image processing unit. The sampling time of the vision system is about 0.1 s, but it can change depending on, e.g., the complexity of the scene, for example, in city traffic (Eidehall 2004).

Lane markings are detected by using differences in the image contrast (Technology Daimler and Safety Innovation 2013). The camera can be either monochrome or full colored. The latter is used to enhance the detection of lane markings, which have different colors around the world (News Releases DENSO Corporation 2013b). Distances to the lane markings and road geometry parameters, like heading angle and curvature, are determined by the image processing algorithms, which must be robust to poor image due to bad weather conditions or worn lane markings. Estimation of road geometry parameters, like curvature measurement, can be a challenging problem (Lundquist and Schön 2011), especially during rain or fog (Eidehall 2004).

Depending on the image processing algorithms the cameras are equipped with, surrounding objects can also be detected and tracked. In particular, pattern recognition algorithm can be used to find objects in the images and classify them into cars, trucks, motorcycles, and pedestrians. Vehicles (or other objects) can be typically detected in a range of about 60–70 m, with lower accuracy than a radar (Eidehall 2004).

Actuators

In order to keep the vehicle within its lane, the most convenient actuator is the steering. Hence, a lane keeping system can be quite easily built in those vehicles equipped with electric power-assisted steering (EPAS) systems. In particular, an additional steering torque can be added by the EPAS to the driver's steering torque, in order to generate the desired yaw moment calculated by the *decision-making and control module*.

Clearly, the steering command is not the only available to affect the vehicle yaw motion, thus changing its orientation and lateral position within the lane. Individual wheel braking may also be used (Technology Daimler and Safety Innovation 2013). In particular, in vehicles equipped with yaw motion control system via individual braking, a braking torque request for each wheel can be sent to the yaw motion control system in order to generate the desired yaw motion.

Decision Making and Control

The decision making and control in a lane keeping problem can be conceptually divided into two tasks: *the threat assessment* and *the lane position control*. The threat assessment problem can be stated as the problem of detecting the risk of accident due to an unintended lane departure, for a given situation of the surrounding environment (i.e., surrounding vehicles and obstacles). The lane position control problem is the problem of controlling the vehicle yaw and lateral motion in order to stay within the lane. The lane position control is activated once the threat assessment detects the risk of accident.

We point out that the border between the corresponding modules executing these two tasks may be blurred for different existing commercial lane keeping systems. That is, the two problems may not be solved by two separate modules, but rather seen and solved as a single problem. Moreover, the following presentation of the threat assessment and the lane position control problems and approaches abstracts from the implementation of a particular lane keeping system available on the market, rather focusing on fundamental concepts.

Threat Assessment

The core information in a threat assessment algorithm for lane keeping applications is given by a measure called time to lane crossing (TLC). This is the predicted time when a front tire intersects a

lane boundary. As explained in van Winsum et al. (2000), the TLC can be calculated in different ways. Next, its simplest expression is reported as (Eidehall 2004)

$$\text{TLC} = \frac{W/2 - W_{\text{veh}}/2 - y_{\text{off}}}{\dot{y}_{\text{off}}}, \qquad (1)$$

where W is the lane width, y_{off} is the vehicle lateral position within the lane, and W_{veh} is the vehicle width. Equation (1) can be easily modified to calculate the TLC w.r.t. any lane boundary relative to the adjacent lanes.

The simplest way of using the TLC is just monitoring it and triggering an action as the TLC passes a threshold. Nevertheless, depending on the vehicle manufacturer, more sophisticated logics can be developed in order to correctly interpret the driver's intention and minimize the unnecessary assisting interventions. Next, few scenarios follow that must be taken into account while developing such logics in order to not interfere with the driver. In particular, the threat assessment module should stop or not trigger any assisting intervention while the vehicle is approaching or crossing a lane boundary if

- The indicators are active,
- A risk of collision with the vehicle ahead is detected, such that the vehicle is crossing the lane markings as results of an evasive maneuver,
- The radar detects a slower vehicle ahead and the driver accelerates, since this may be an overtaking (Technology Daimler and Safety Innovation 2013),
- The driver's steering wheel torque indicates that the driver is acting against the system,
- The driver manually initiates a maneuver, driving the vehicle back to its lane (i.e., the driver executes "the right" maneuver)
- The vehicle enters a motor highway or a bend (Technology Daimler and Safety Innovation 2013).

Part of the threat assessment task is predicting the trajectories of the surrounding vehicles. For instance, if a *threat* vehicle is traveling in the adjacent lane (in the same or opposite direction), its position has to be predicted at the TLC in order to decide whether to trigger an intervention, if a collision is predicted, or not (Eidehall 2004). This step is repeated for all the detected threat vehicles, provided that the onboard radar and the camera support multiple-target tracking.

In order to minimize the interference of the lane keeping system with the driver and/or to not let the system perform dangerous maneuvers, assisting interventions should not be triggered if the quality of the measurements is such that the information about the surrounding environment is poor. For instance, in case of low visibility that limits the detection of the lane markings and the estimation of the road geometry, the system should be temporarily deactivated or downgraded.

In summary, the threat assessment module has to be designed with the objective of detecting the risk of accident due to lane departure while not interfering with the driver with unnecessary interventions (i.e., nuisance minimization).

Lane Position Control

As observed in section "Actuators," the vehicle motion within the lane can be affected in two ways, i.e., through steering and individual wheel braking. Clearly, a steering command can be issued by both the driver and the lane keeping system.

Before issuing a steering command, in order to minimize the system nuisance, the lane keeping system may issue other types of low-intrusiveness interventions. For instance, if a "low"-level threat is detected by the threat assessment module (i.e., a threat where the risk of accidents is not imminent), warnings or other stimuli to the driver may be issued in order to induce the driver to execute the right maneuver. For instance, based on, e.g., spectrum analysis of the driver's steering command, driver's inattention or drowsiness may be detected and a warning issued. As observed in Technology Daimler and Safety Innovation (2013), different types of warning can be used for different vehicle types. In passenger cars, in such cases, a vibration motor in the steering wheel may warn the driver. In trucks, audible, directional warning signals can be used to let the driver know
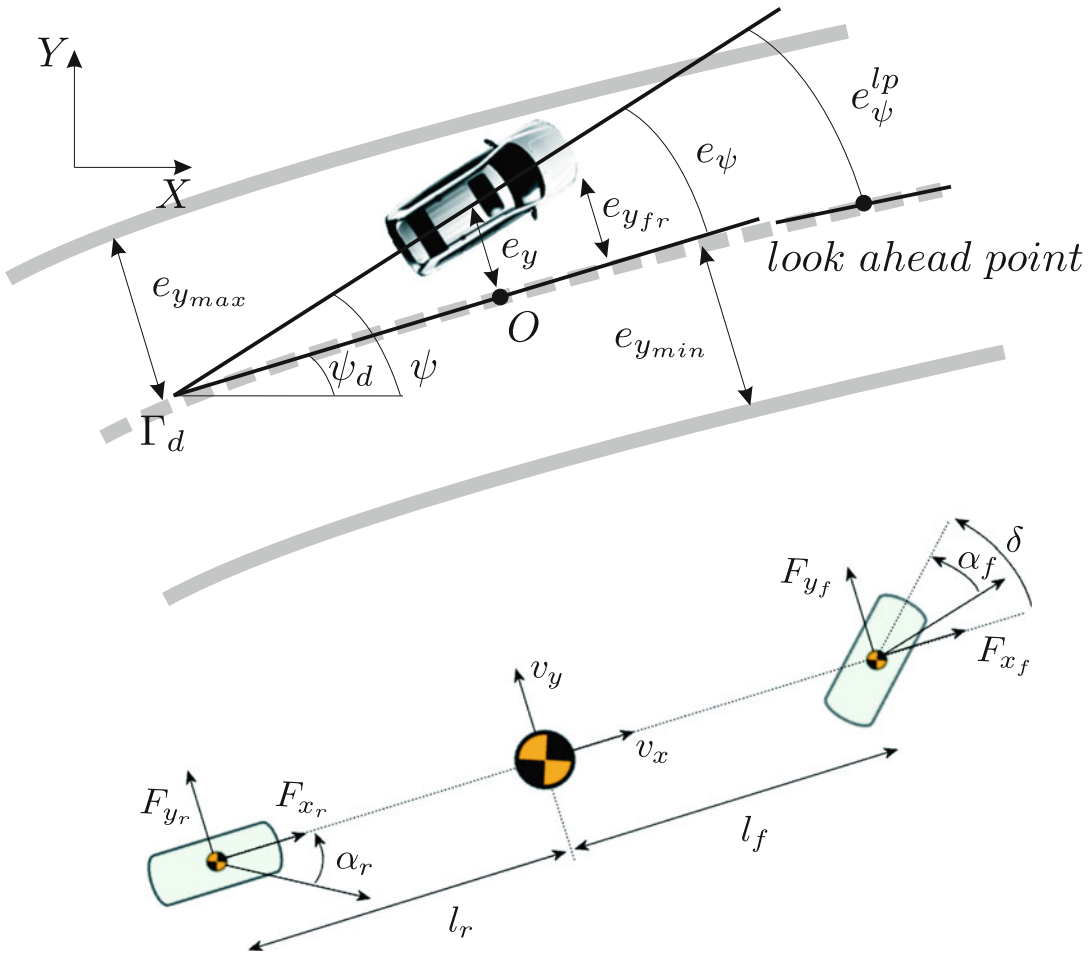
Lane Keeping, Fig. 2 Vehicle modeling notation

that the vehicle trajectory needs to be adjusted. In buses, in order to avoid bothering the passengers, driver warning is issued through vibration motors placed in the driver's seat.

Other types of "soft intervention" aim at increasing the steering impedance in the direction leading to lane crossing that might cause a collision with surrounding vehicles. Generating the desired steering impedance can be easily formulated as a steering torque control problem. Nevertheless, tuning the control algorithm to obtain the desired steering feeling can be an involving and time-consuming procedure based on extensive in-vehicle testing.

Besides warnings and "soft interventions" aiming at inducing the driver to perform correct maneuvers, as part of the lane position control

task in a lane keeping system, a lateral control algorithm w.r.t. the lane boundaries is needed. Consider the vehicle sketched in Fig. 2. The equations describing the vehicle motion within the lane can be compactly written in a state-space form as

$$\dot{x} = Ax + B\delta + D\dot{\psi}_{\text{des}}, \qquad (2)$$

where $x = \begin{bmatrix} e_y & \dot{e}_y & e_\psi & \dot{e}_\psi \end{bmatrix}$, $\dot{\psi}_d$ is the desired yaw rate, e.g., calculated based on the road curvature, and A, B, D are speed-dependent matrices that can be found in Rajamani (2003). The (unstable) system can be stabilized by a state-feedback control law

$$\delta = -Kx + \delta_{ff}, \qquad (3)$$

where K is a stabilizing static gain and δ_{ff} is a feedforward term that can be used to compensate for the road curvature. In Rajamani (2003), it is shown that, while $e_y(t) \to 0$ as $t \to 0$, e_ψ approaches a nonzero steady-state value, no matter how δ_{ff} is chosen, for non-straight road.

Despite a simple problem formulation and solution, controlling the vehicle position within the lane is not a trivial task. Indeed, having the control law (3) active all the time may increase the nuisance, leading to unacceptable driving experience. For this reason, the steering command calculated through the (3) may be active only when the vehicle significantly deviates from the road centerline, i.e., approaches the lane markings. Clearly, adding such logics complicates the analysis of the closed-loop behavior, thus making necessary extensive in-vehicle tuning and verification.

Summary and Future Directions

In this chapter, we have overviewed the general issues and requirements that must be considered in the design of a lane-keeping system.

The variety of environmental conditions the sensing system should operate in, together with the range of diverse scenarios the decision-making module should cope with, render the design and verification problems challenging, costly, and time consuming for a lane-keeping system. It is, therefore, necessary to approach the design of such systems by also providing safety guarantees to the largest extent, yet minimizing conservatism and intrusiveness of the overall system. Model-based approaches to threat assessment and decision-making problems, as proposed in Falcone et al. (2011) for a lane departure application, provide neat design and verification frameworks, which can clearly describe the safe operation of the overall system. Adopting such design methodologies can potentially contribute to a consistent reduction of the development time by consistently reducing the a posteriori safety verification phase. On the other hand, the computational complexity of formal model-based verification methods can dramatically increase in those scenarios where system nonlinearity and nonconvex state spaces become relevant. Hence, future research efforts aiming at developing low-complexity verification methods might greatly impact the future development of automated driving systems.

Cross-References

▶ Adaptive Cruise Control
▶ Vehicle Dynamics Control

Bibliography

Eidehall A (2004) An automotive lane guidance system. Lic thesis 1122, Linköping University

Falcone P, Ali M, Sjöberg J (2011) Predictive threat assessment via reachability analysis and set invariance theory. IEEE Trans Intell Transp Syst 12(4):1352–1361

Lundquist C, Schön TB (2011) Joint ego-motion and road geometry estimation. Inf Fusion 4(12):253–263

News Releases DENSO Corporation (2013a) Denso develops higher performance millimeter-wave radar, June 2013

News Releases DENSO Corporation (2013b) Denso develops new vision sensor for active safety systems, June 2013

Rajamani R (2006) Vehicle dynamics and control, chap. 2. Springer, New York

Technology Daimler and Safety Innovation (2013) Lane keeping assist: always on the right track, June 2013

van Winsum W, Brookhuis KA, de Waard D (2000) A comparison of different ways to approximate time-to-line crossing (TLC) during car driving. Accid Anal Prev 32(1):47–56

Learning in Games

Jeff S. Shamma
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

In a Nash equilibrium, each player selects a strategy that is optimal with respect to the strategies of other players. This definition does not mention the process by which players reach a

Nash equilibrium. The topic of learning in games seeks to address this issue in that it explores how simplistic learning/adaptation rules can lead to Nash equilibrium. This entry presents a selective sampling of learning rules and their long-run convergence properties, i.e., conditions under which player strategies converge or not to Nash equilibrium.

Keywords

Cournot best response; Fictitious play; Log-linear learning; Mixed strategies; Nash equilibrium

Introduction

In a *Nash equilibrium*, each player's strategy is optimal with respect to the strategies of other players. Accordingly, Nash equilibrium offers a predictive model of the outcome of a game. That is, given the basic elements of a game – (i) a set of players; (ii) for each player, a set of strategies; and (iii) for each player, a utility function that captures preferences over strategies – one can model/assert that the strategies selected by the players constitute a Nash equilibrium.

In making this assertion, there is no suggestion of *how* players may come to reach a Nash equilibrium. Two motivating quotations in this regard are:

> The attainment of equilibrium requires a disequilibrium process (Arrow 1986).

and

> The explanatory significance of the equilibrium concept depends on the underlying dynamics (Skyrms 1992).

These quotations reflect that a foundation for Nash equilibrium as a predictive model is dynamics that lead to equilibrium. Motivated by these considerations, the topic of "learning in games" shifts the attention away from equilibrium and towards underlying dynamic processes and their long-run behavior. The intent is to understand how players may reach an equilibrium as well

as understand possible barriers to reaching Nash equilibrium.

In the setup of learning in games, players repetitively play a game over a sequence of stages. At each stage, players use past experiences/observations to select a strategy for the current stage. Once player strategies are selected, the game is played, information is updated, and the process is repeated. The question is then to understand the long-run behavior, e.g., whether or not player strategies converge to Nash equilibrium.

Traditionally the dynamic processes considered under learning in games have players selecting strategies based on a myopic desire to optimize for the current stage. That is, players do not consider long-run effects in updating their strategies. Accordingly, while players are engaged in repetitive play, the dynamic processes generally are not optimal in the long run (as in the setting of "repeated games"). Indeed, the survey article of Hart (2005) refers to the dynamic processes of learning in games as "adaptive heuristics." This distinction is important in that an implicit concern in learning in games is to understand how "low rationality" (i.e., suboptimal and heuristic) processes can lead to the "high rationality" (i.e., mutually optimal) notion of Nash equilibrium.

This entry presents a sampling of results from the learning in games literature through a selection of illustrative dynamic processes, a review of their long-run behaviors relevant to Nash equilibrium, and pointers to further work.

Illustration: Commuting Game

We begin with a description of learning in games in the specific setting of the commuting game, which is a special case of so-called congestion games (cf., Roughgarden 2005). The setup is as follows. Each player seeks to plan a path from an origin to a destination. The origins and destinations can differ from player to player. Players seek to minimize their own travel times. These travel times depend both on the chosen path (distance traveled) *and* the paths of other players (road congestion). Every day, a player uses past

information and observations to select that day's path according to some selection rule, and this process is repeated day after day.

In game-theoretic terms, player "strategies" are paths linking their origins to destinations, and player "utility functions" reflect travel times. At a Nash equilibrium, players have selected paths such that no individual player can find a shorter travel time given the chosen paths of others. The learning in games question is then whether player paths indeed converge to Nash equilibrium in the long run. Not surprisingly, the answer depends on the specific process that players use to select paths and possible additional structure of the commuting game.

Suppose that one of the players, say "Alice," is choosing among a collection of paths. For the sake of illustration, let us give Alice the following capabilities: (i) Alice can observe the paths chosen by all other players and (ii) Alice can compute off-line her travel time as a function of her path and the paths of others.

With these capabilities, Alice can compute running averages of the travel times along all available paths. Note that the assumed capabilities allow Alice to compute the travel time of a path and hence its running average, *whether or not* she took the path on that day. With average travel time values in hand, two possible learning rules are:

– *Exploitation:* Choose the path with the lowest average travel time.
– *Exploitation with Exploration:* With high probability, choose the path with the lowest average travel time, and with low probability, choose a path at random.

Assuming that all players implement the same learning rule, each case induces a dynamic process that governs the daily selection of paths and determines the resulting long-run behavior. We will revisit these processes in a more formal setting in the next section.

A noteworthy feature of these learning rules is that they do not explicitly depend on the utility functions of other players. For example, suppose one of the other players is willing to trade off travel time for more scenic routes. Similarly, suppose one of the other players prefers to travel

on high congestion paths, e.g., a rolling billboard seeking to maximize exposure. The aforementioned learning rules for Alice remain unchanged. Of course, Alice's actions implicitly depend on the utility functions of other players, but only indirectly through their selected paths. This characteristic of no explicit dependence on the utility functions of others is known as "uncoupled" learning, and it can have major implications on the achievable long-run behavior (Hart and Mas-Colell 2003a).

In assuming the ability to observe the paths of other players and to compute off-line travel times as a function of these paths, these learning rules impose severe requirements on the information available to each player. Less restrictive are learning rules that are "payoff based" (Young 2005). A simple modification that leads to payoff-based learning is as follows. Alice maintains an empirical average of the travel times of a path using only the days that she took that path. Note the distinction – on any given day, Alice remains unaware of travel times for the routes not selected. Using these empirical average travel times, Alice can then mimic any of the aforementioned learning rules. As intended, she does not directly observe the paths of others, nor does she have a closed-form expression for travel times as a function of player paths. Rather, she only can select a path and measure the consequences. As before, all players implementing such a learning rule induce a dynamic process, but the ensuing analysis in payoff-based learning can be more subtle.

Learning Dynamics

We now give a more formal presentation of selected learning rules and results concerning their long-run behavior.

Preliminaries

We begin with the basic setup of games with a finite set of players, $\{1, 2, \ldots, N\}$, and for each player i, a finite set of strategies, $\mathcal{A}_i$. Let

$$\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_N$$

denote the set of strategy profiles. Each player, i, is endowed with a utility function

$$u_i : \mathcal{A} \to \mathbb{R}.$$

Utility functions capture player preferences over strategy profiles. Accordingly, for any $a, a' \in \mathcal{A}$, the condition

$$u_i(a) > u_i(a')$$

indicates that player i prefers the strategy profile a over a'.

The notation $-i$ indicates the set of players *other than* player i. Accordingly, we sometimes write $a \in \mathcal{A}$ as (a_i, a_{-i}) to isolate a_i, the strategy of player i, versus a_{-i}, the strategies of other players. The notation $-i$ is used in other settings as well.

Utility functions induce best-response sets. For $a_{-i} \in \mathcal{A}_{-i}$, define

$$\mathcal{B}_i(a_{-i}) = \{ a_i : u_i(a_i, a_{-i}) \ge u_i(a'_i, a_{-i})$$
$$\text{for all } a'_i \in \mathcal{A}_i \}.$$

In words, $\mathcal{B}_i(a_{-i})$ denotes the set of strategies that are optimal for player i in response to the strategies of other players, a_{-i}.

A strategy profile $a^* \in \mathcal{A}$ is a *Nash equilibrium* if for any player i and any $a'_i \in \mathcal{A}_i$,

$$u_i(a^*_i, a^*_{-i}) \ge u_i(a'_i, a^*_{-i}).$$

In words, at a Nash equilibrium, no player can achieve greater utility by unilaterally changing strategies. Stated in terms of best-response sets, a strategy profile, a^*, is a Nash equilibrium if for every player i,

$$a^*_i \in \mathcal{B}_i(a^*_{-i}).$$

We also will need the notions of *mixed strategies* and mixed strategy Nash equilibrium. Let $\Delta(\mathcal{A}_i)$ denote probability distributions (i.e., nonnegative vectors that sum to one) over the set $\mathcal{A}_i$. A mixed strategy profile is a collection of probability distributions, $\alpha = (\alpha_1, \ldots, \alpha_N)$, with

$\alpha_i \in \Delta(\mathcal{A}_i)$ for each i. Let us assume that players choose a strategy randomly and independently according to these mixed strategies. Accordingly, define $\mathbf{Pr}[a; \alpha]$ to be the probability of strategy a under the mixed strategy profile α, and define the expected utility of player i as

$$U_i(\alpha) = \sum_{a \in \mathcal{A}} u_i(a) \cdot \mathbf{Pr}[a; \alpha].$$

A *mixed strategy Nash equilibrium* is a mixed strategy profile, α^*, such that for any player i and any $\alpha'_i \in \Delta(\mathcal{A})$,

$$U_i(\alpha^*_i, \alpha^*_{-i}) \ge U_i(\alpha'_i, \alpha^*_{-i}).$$

Special Classes of Games

We will reference three special classes of games: (i) zero-sum games, (ii) potential games, and (iii) weakly acyclic games.

Zero-sum games: There are only two players (i.e., $N = 2$), and $u_1(a) = -u_2(a)$.

Potential games: There exists a (potential) function,

$$\phi : \mathcal{A} \to \mathbb{R}$$

such that for any pair of strategies, $a = (a_i, a_{-i})$ and $a' = (a'_i, a_{-i})$, that differ only in the strategy of player i,

$$u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i}) = \phi(a_i, a_{-i}) - \phi(a'_i, a_{-i}).$$

Weakly acyclic games: There exists a function

$$\phi : \mathcal{A} \to \mathbb{R}$$

with the following property: if $a \in \mathcal{A}$ is *not* a Nash equilibrium, then at least one player, say player i, has an alternative strategy, say $a'_i \in \mathcal{A}_i$, such that

$$u_i(a'_i, a_{-i}) > u_i(a_i, a_{-i})$$

and

$$\phi(a'_i, a_{-i}) > \phi(a_i, a_{-i}).$$

Potential games are a special class of games for which various learning dynamics converge to

a Nash equilibrium. The aforementioned commuting game constitutes a potential game under certain special assumptions. These are as follows: (i) the delay on a road only depends on the number of users (and not their identities) and (ii) all players measure delay in the same manner (Monderer and Shapley 1996).

Weakly acyclic games are a generalization of potential games. In potential games, there exists a potential function that captures differences in utility under unilateral (i.e., single player) changes in strategy. In weakly acyclic games (see Young 1998), if a strategy profile is not a Nash equilibrium, then there *exists* a player who can simultaneously achieve an increase in utility while increasing the potential function. The characterization of weakly acyclic games through a potential function herein is not traditional and is borrowed from Marden et al. (2009a).

Forecasted Best-Response Dynamics

One family of learning dynamics involves players formulating a forecast of the strategies of other players based on past observations and then playing a best response to this forecast.

Cournot Best-Response Dynamics

The simplest illustration is Cournot best-response dynamics. Players repetitively play the same game over stages $t = 0, 1, 2, \ldots$. At stage t, a player forecasts that the strategies of other players are the strategies played at the previous stage $t - 1$. The following rules specify Cournot best response with inertia. For each stage t and for each player i:

- With probability $p \in (0, 1)$, $a_i(t) = a_i(t - 1)$ (inertia).
- With probability $1 - p$, $a_i(t) \in \mathcal{B}_i(a_{-i}(t-1))$ (best response).
- If $a_i(t - 1) \in \mathcal{B}_i(a_{-i}(t - 1))$, then $a_i(t) = a_i(t - 1)$ (continuation).

Proposition 1 *For weakly acyclic (and hence potential) games, player strategies under Cournot best-response dynamics with inertia converge to a Nash equilibrium.*

Cournot best-response dynamics need not always converge in games with a Nash equilibrium, hence the restriction to weakly acyclic games.

Fictitious Play

In fictitious play, introduced in Brown (1951), players also use past observations to construct a forecast of the strategies of other players. Unlike Cournot best-response dynamics, this forecast is *probabilistic*.

As a simple example, consider the commuting game with two players, Alice and Bob, who both must choose between two paths, A and B. Now suppose that on stage $t = 10$, Alice has observed Bob used path A for 6 out of the previous 10 days and path B for the remaining days. Then Alice's forecast of Bob is that he will chose path A with 60 % probability and path B with 40 % probability. Alice then chooses between path A and B in order to optimize her *expected* utility. Likewise, Bob uses Alice's empirical averages to form a probabilistic forecast of her next choice and selects a path to optimize his expected utility.

More generally, let $\pi_j(t) \in \Delta(\mathcal{A}_j)$ denote the *empirical frequency* for player j at stage t. This vector is a probability distribution that indicates the relative frequency of times player j played each strategy in $\mathcal{A}_j$ over stages $0, 1, \ldots, t - 1$. In fictitious play, player i assumes (incorrectly) that at stage t, other players will select their strategies independently and randomly according to their empirical frequency vectors. Let $\Pi_{-i}(t)$ denote the induced probability distribution over $\mathcal{A}_{-i}$ at stage t. Under fictitious play, player i selects an action according to

$$a_i(t) \in \arg \max_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} u_i(a_i, a_{-i})$$

$$\cdot \mathbf{Pr}[a_{-i}; \Pi_{-i}(t)].$$

In words, player i selects the action that maximizes expected utility assuming that other players select their strategies randomly and independently according to their empirical frequencies.

Proposition 2 *For (i) zero-sum games, (ii) potential games, and (iii) two-player games*

in which one player has only two actions, player empirical frequencies under fictitious play converge to a mixed strategy Nash equilibrium.

These results are reported in Fudenberg and Levine (1998), Hofbauer and Sandholm (2002), and Berger (2005). Fictitious play need not converge to Nash equilibria in all games. An early counterexample is reported in Shapley (1964), which constructs a two-player game with a unique mixed strategy Nash equilibrium. A weakly acyclic game with multiple pure (i.e., non-mixed) Nash equilibria under which fictitious play does not converge is reported in Foster and Young (1998).

A variant of fictitious play is "joint strategy" fictitious play (Marden et al. 2009b). In this framework, players construct as forecasts empirical frequencies of the *joint* play of other players. This formulation is in contrast to constructing and combining empirical frequencies for each player. In the commuting game, it turns out that joint strategy fictitious play is equivalent to the aforementioned "exploitation" rule of selecting the path with lowest average travel time. Marden et al. (2009b) show that action profiles under joint strategy fictitious play (with inertia) converge to a Nash equilibrium in potential games.

Log-Linear Learning

Under forecasted best-response dynamics, players chose a best response to the forecasted strategies of other players. Log-linear learning, introduced in Blume (1993), allows the possibility of "exploration," in which players can select nonoptimal strategies but with relatively low probabilities.

Log-linear learning proceeds as follows. First, introduce a "temperature" parameter, $T > 0$.

– At stage t, a single player, say player i, is selected at random.
– For player i,

$$\mathbf{Pr}[a_i(t) = a_i'] = \frac{1}{Z} e^{u_i(a_i', a_{-i}(t-1))/T}.$$

– For all other players, $j \neq i$,

$$a_j(t) = a_j(t - 1).$$

In words, under log-linear learning, only a single player performs a strategy update at each stage. The probability of selecting a strategy is exponentially proportional to the utility garnered from that strategy (with other players repeating their previous strategies). In the above description, the dummy parameter Z is a normalizing variable used to define a probability distribution. In fact, the specific probability distribution for strategy selection is a Gibbs distribution with temperature parameter, T. For very large T, strategies are chosen approximately uniformly at random. However, for small T, the selected strategy is a best response (i.e., $a_i(t) \in \mathcal{B}_i(a_{-i}(t-1))$) with high probability, and an alternative strategy is selected with low probability.

Because of the inherent randomness, strategy profiles under log-linear learning never converge. Nonetheless, the long-run behavior can be characterized probabilistically as follows.

Proposition 3 *For potential games with potential function $\phi(\cdot)$ under log-linear learning, for any $a \in \mathcal{A}$,*

$$\lim_{t \to \infty} \mathbf{Pr}[a(t) = a] = \frac{1}{Z} e^{\phi(a)/T}.$$

In words, the long-run probabilities of strategy profiles conform to a Gibbs distribution constructed from the underlying potential function. This characterization has the important implication of (probabilistic) equilibrium *selection*. Prior convergence results stated convergence to Nash equilibria, but did not specify which Nash equilibrium in the case of multiple equilibria. Under log-linear learning, there is a probabilistic preference for the Nash equilibrium that maximizes the underlying potential function.

Extensions and Variations

Payoff-based learning. The discussion herein presumed that players can observe the actions of other players and can compute utility functions off-line. Payoff-based algorithms, i.e., algorithms in which players only measure the utility

garnered in each stage, impose less restrictive informational requirements. See Young (2005) for a general discussion, as well as Marden et al. (2009c), Marden and Shamma (2012), and Arslan and Shamma (2004) for various payoff-based extensions.

No-regret learning. The broad class of so-called "no-regret" learning rules has the desirable property of converging to broader solution concepts (namely, Hannan consistency sets and correlated equilibria) in general games. See Hart and Mas-Colell (2000, 2001, 2003b) for an extensive discussion.

Calibrated forecasts. Calibrated forecasts are more sophisticated than empirical frequencies in that they satisfy certain long-run consistency properties. Accordingly, forecasted best-response learning using calibrated forecasts has stronger guaranteed convergence properties, such as convergence to correlated equilibria. See Foster and Vohra (1997), Kakade and Foster (2008), and Mannor et al. (2007).

Impossibility results. This entry focused on convergence results in various special cases. There are broad impossibility results that imply the impossibility of families of learning rules to converge to Nash equilibria in all games. The focus is on *uncoupled* learning, i.e., the learning dynamics for player i does not depend explicitly on the utility functions of other players (which is satisfied by all of the learning dynamics presented herein). See Hart and Mas-Colell (2003a, 2006), Hart and Mansour (2007), and Shamma and Arslan (2005). Another type of impossibility result concerns lower bounds on the required rate of convergence to equilibrium (e.g., Hart and Mansour 2010).

Welfare maximization. Of special interest is learning dynamics that select welfare (i.e., sum of utilities) maximizing strategy profiles, whether or not they are Nash equilibria. Recent contributions include Pradelski and Young (2012), Marden et al. (2011), and Arieli and Babichenko (2012).

Summary and Future Directions

We have presented a selection of learning dynamics and their long-run characteristics, specifically in terms of convergence to Nash equilibria. As stated early on, the original motivation of learning in games research has been to add credence to solution concepts such as Nash equilibrium as a model of the outcome of a game. An emerging line of research stems from engineering considerations, in which the objective is to use the framework of learning in games as a design tool for distributed decision architecture settings such as autonomous vehicle teams, communication networks, or smart grid energy systems. A related emerging direction is social influence, in which the objective is to steer the collective behaviors of human decision makers towards a socially desirable situation through the dispersement of incentives. Accordingly, learning in games can offer baseline models on how individuals update their behaviors to guide and inform social influence policies.

Cross-References

▶ Evolutionary Games
▶ Stochastic Games and Learning

Recommended Reading

Monographs on learning in games:

– Fudenberg D, Levine DK (1998) The theory of learning in games. MIT, Cambridge
– Hart S, Mas Colell A (2013) Simple adaptive strategies: from regret-matching to uncoupled dynamics. World Scientific Publishing Company
– Young HP (1998) Individual strategy and social structure. Princeton University Press, Princeton
– Young HP (2005) Strategic learning and its limits. Princeton University Press, Princeton

Overview articles on learning in games:

– Fudenberg D, Levine DK (2009) Learning and equilibrium. Annu Rev Econ 1:385–420

– Hart S (2005) Adaptive heuristics. Econometrica 73(5):1401–1430
– Young HP (2007) The possible and the impossible in multi-agent learning. Artif Intell 171:429–433

Articles relating learning in games to distributed control:

– Li N, Marden JR (2013) Designing games for distributed optimization. IEEE J Sel Top Signal Process 7:230–242
– Mannor S, Shamma JS (2007) Multi-agent learning for engineers, Artif Intell 171:417–422
– Marden JR, Arslan G, Shamma JS (2009) Cooperative control and potential games. IEEE Trans Syst Man Cybern B Cybern 6:1393–1407

Bibliography

Arieli I, Babichenko Y (2012) Average-testing and Pareto efficiency. J Econ Theory 147: 2376–2398
Arrow KJ (1986) Rationality of self and others in an economic system. J Bus 59(4):S385–S399
Arslan G, Shamma JS (2004) Distributed convergence to Nash equilibria with local utility measurements. In: Proceedings of the 43rd IEEE conference on decision and control. Atlantis, Paradise Island, Bahamas
Berger U (2005) Fictitious play in $2 \times n$ games. J Econ Theory 120(2):139–154
Blume L (1993) The statistical mechanics of strategic interaction. Games Econ Behav 5:387–424
Brown GW (1951) Iterative solutions of games by fictitious play. In: Koopmans TC (ed) Activity analysis of production and allocation. Wiley, New York, pp 374–376
Foster DP, Vohra R (1997) Calibrated learning and correlated equilibrium. Games Econ Behav 21:40–55
Foster DP, Young HP (1998) On the nonconvergence of fictitious play in coordination games. Games Econ Behav 25:79–96
Fudenberg D, Levine DK (1998) The theory of learning in games. MIT, Cambridge
Hart S (2005) Adaptive heuristics. Econometrica 73(5):1401–1430
Hart S, Mansour Y (2007) The communication complexity of uncoupled Nash equilibrium procedures. In: STOC'07 proceedings of the 39th annual ACM symposium on the theory of computing, San Diego, CA, USA, pp 345–353
Hart S, Mansour Y (2010) How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. Games Econ Behav 69(1):107–126

Hart S, Mas-Colell A (2000) A simple adaptive procedure leading to correlated equilibrium. Econometrica 68(5):1127–1150
Hart S, Mas-Colell A (2001) A general class of adaptive strategies. J Econ Theory 98:26–54
Hart S, Mas-Colell A (2003a) Uncoupled dynamics do not lead to Nash equilibrium. Am Econ Rev 93(5):1830–1836
Hart S, Mas-Colell A (2003b) Regret based continuous-time dynamics. Games Econ Behav 45:375–394
Hart S, Mas-Colell A (2006) Stochastic uncoupled dynamics and Nash equilibrium. Games Econ Behav 57(2):286–303
Hofbauer J, Sandholm W (2002) On the global convergence of stochastic fictitious play. Econometrica 70:2265–2294
Kakade SM, Foster DP (2008) Deterministic calibration and Nash equilibrium. J Comput Syst Sci 74: 115–130
Mannor S, Shamma JS, Arslan G (2007) Online calibrated forecasts: memory efficiency vs universality for learning in games. Mach Learn 67(1):77–115
Marden JR, Shamma JS (2012) Revisiting log-linear learning: asynchrony, completeness and a payoff-based implementation. Games Econ Behav 75(2):788–808
Marden JR, Arslan G, Shamma JS (2009a) Cooperative control and potential games. IEEE Trans Syst Man Cybern Part B Cybern 39:1393–1407
Marden JR, Arslan G, Shamma JS (2009b) Joint strategy fictitious play with inertia for potential games. IEEE Trans Autom Control 54:208–220
Marden JR, Young HP, Arslan G, Shamma JS (2009c) Payoff based dynamics for multi-player weakly acyclic games. SIAM J Control Optim 48:373–396
Marden JR, Peyton Young H, Pao LY (2011) Achieving Pareto optimality through distributed learning. Oxford economics discussion paper #557
Monderer D, Shapley LS (1996) Potential games. Games Econ Behav 14:124–143
Pradelski BR, Young HP (2012) Learning efficient Nash equilibria in distributed systems. Games Econ Behav 75:882–897
Roughgarden T (2005) Selfish routing and the price of anarchy. MIT, Cambridge
Shamma JS, Arslan G (2005) Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria. IEEE Trans Autom Control 50(3):312–327
Shapley LS (1964) Some topics in two-person games. In: Dresher M, Shapley LS, Tucker AW (eds) Advances in game theory. University Press, Princeton, pp 1–29
Skyrms B (1992) Chaos and the explanatory significance of equilibrium: strange attractors in evolutionary game dynamics. In: PSA: proceedings of the biennial meeting of the Philosophy of Science Association. Volume Two: symposia and invited papers, East Lansing, MI, USA, pp 274–394
Young HP (1998) Individual strategy and social structure. Princeton University Press, Princeton
Young HP (2005) Strategic learning and its limits. Oxford University Press. Oxford

L

Learning Theory

Mathukumalli Vidyasagar
University of Texas at Dallas, Richardson,
TX, USA

Introduction

How does a machine learn an abstract concept from examples? How can a machine generalize to previously unseen situations? Learning theory is the study of (formalized versions of) such questions. There are many possible ways to formulate such questions. Therefore, the focus of this entry is on one particular formalism, known as PAC (probably approximately correct) learning. It turns out that PAC learning theory is rich enough to capture intuitive notions of what learning should mean in the context of applications and, at the same time, is amenable to formal mathematical analysis. There are several precise and complete studies of PAC learning theory, many of which are cited in the bibliography. Therefore, this article is devoted to sketching some high-level ideas.

Keywords

Machine learning; Probably approximately correct (PAC) learning; Support vector machine; Vapnik-Chervonenkis (V-C) dimension

Problem Formulation

In the PAC formalism, the starting point is the premise that there is an unknown set, say an unknown convex polygon, or an unknown half-plane. The unknown set cannot be *completely* unknown; rather, something should be specified about its nature, in order for the problem to be both meaningful and tractable. For instance, in the first example above, the learner knows that the unknown set is a convex polygon, though it is not known *which* polygon it might be.

Similarly, in the second example, the learner knows that the unknown set is a half-plane, though it is not known *which* half-plane. The collection of all possible unknown sets is known as the **concept class**, and the particular unknown set is referred to as the "target concept." In the first example, this would be the set of all convex polygons and in the second case it would be the set of half-planes. The unknown set cannot be directly observed of course; otherwise, there would be nothing to learn. Rather, one is given clues about the target concept by an "oracle," which informs the learner whether or not a particular element belongs to the target concept. Therefore, the information available to the learner is a collection of "labelled samples," in the form $\{(x_i, I_T(x_i), i = 1, \ldots, m\}$, where m is the total number of labelled samples and $I_t(\cdot)$ is the indicator function of the target concept T. Based on this information, the learner is expected to generate a "hypothesis" H_m that is a good approximation to the unknown target concept T.

One of the main features of PAC learning theory that distinguishes it from its forerunners is the observation that, no matter how many training samples are available to the learner, the hypothesis H_m can never *exactly equal* the unknown target concept T. Rather, all that one can expect is that H_m converges to T in some appropriate metric. Since the purpose of machine learning is to generate a hypothesis H_m that can be used to approximate the unknown target concept T for prediction purposes, a natural candidate for the metric that measures the disparity between H_m and T is the so-called generalization error, defined as follows: Suppose that, after m training samples that have led to the hypothesis H_m, a testing sample x is generated at random. One can now ask: what is the probability that the hypothesis H_m misclassifies x? In other words, what is the value of $\Pr\{I_{H_m}(x) \neq I_T(x)\}$? This quantity is known as the generalization error, and the objective is to ensure that it approaches zero as $m \to \infty$.

The manner in which the samples are generated leads to different models of learning. For instance, if the learner is able to choose the next sample x_{m+1} on the basis of the previous

m labelled samples, which is then passed on to the oracle for labeling, this is known as "active learning." More common is "passive learning," in which the sequence of training samples $\{x_i\}_{i \geq 1}$ is generated at random, in an independent and identically distributed (i.i.d.) fashion, according to some probability distribution P. In this case, even the hypothesis H_m and the generalization error are random, because they depend on the randomly generated training samples. This is the rationale behind the nomenclature "probably approximately correct." The hypothesis H_m is not expected to equal to unknown target concept T exactly, only approximately. Even that is only probably true, because in principle it is possible that the randomly generated training samples could be totally unrepresentative and thus lead to a poor hypothesis. If we toss a coin many times, there is a small but always positive probability that it could turn up heads every time. As the coin is tossed more and more times, this probability becomes smaller, but will never equal zero.

Examples

Example 1 Consider the situation where the concept class consists of all half-planes in $\mathbb{R}^2$, as indicated in the left side of Fig. 1. Here the unknown target concept T is some fixed but unknown half-plane. The symbol T is next to the boundary of the half-plane, and all points to the right of the line constitute the target half-plane. The training samples, generated at random according some unknown probability distribution P, are also shown in the figure. The samples that belong to T are shown as blue rectangles, while those that do not belong to T are shown as red dots. Knowing only these labelled samples, the learner is expected to guess what T might be.

A reasonable approach is to choose some half-plane that agrees with the data and correctly classifies the labelled data. For instance, the well-known support vector machine (SVM) algorithm chooses the unique half-plane such that the closest sample to the dividing line is as far as possible from it; see the paper by Cortes and Vapnik (1997).

The symbol H denotes the boundary of a hypothesis, which is another half-plane. The shaded region is the **symmetric difference** between the two half-planes. The set $T \Delta H$ is the set of points that are misclassified by the hypothesis H. Of course, we do not know what this set is, because we do not know T. It can be shown that, whenever the hypothesis H is chosen to be **consistent** in the sense of correctly classifying all labelled samples, the generalization error goes to zero as the number of samples approaches infinity.

Example 2 Now suppose the concept class consists of all convex polygons in the unit square, and let T denote the (unknown) target convex polygon. This situation is depicted in the right side of Fig. 1. This time let us assume that the probability distribution that generates the samples is the uniform distribution on X. Given a set of positively and negatively labelled samples (the same convention as in Example 1), let us choose the hypothesis H to be the convex hull of all positively labelled samples, as shown in the figure. Since every positively labelled sample

Learning Theory, Fig. 1 Examples of learning problems. (**a**) Learning half-planes. (**b**) Learning convex polygons
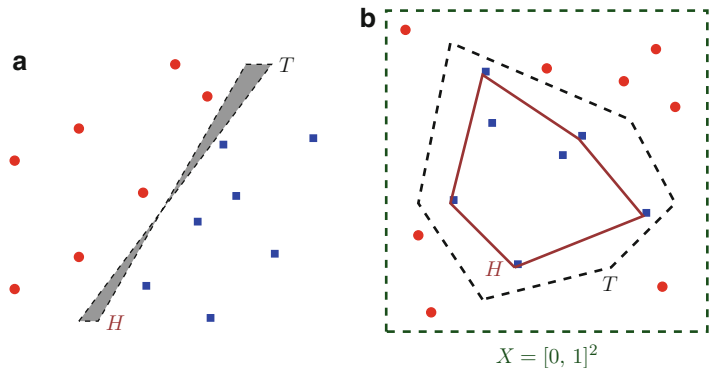
Learning Theory, Fig. 2
VC dimension illustrations.
(**a**) Shattering a set of three
elements. (**b**) Infinite VC
dimension
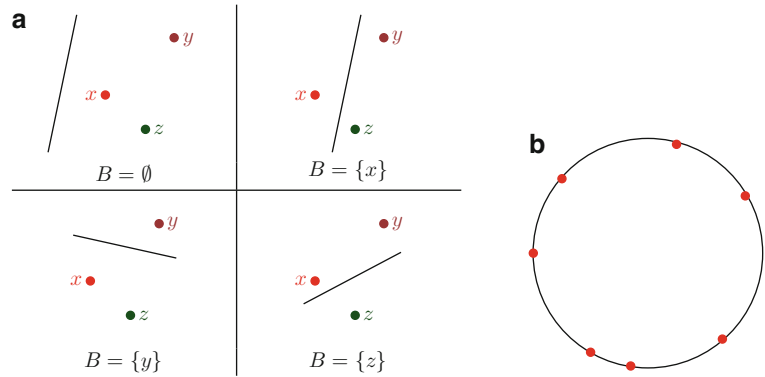
belongs to T, and T is a convex set, it follows
that H is a subset of T. Moreover, $P(T \setminus H)$
is the generalization error. It can be shown that
this algorithm also "works" in the sense that the
generalization error goes to zero as the number of
samples approaches infinity.

Vapnik-Chervonenkis Dimension

Given any concept class $\mathcal{C}$, there is a single
integer that offers a measure of the richness of
the class, known as the Vapnik-Chervonenkis (or
VC) dimension, after its originators.

Definition 1 A set $S \subseteq X$ is said to be **shattered**
by a concept class $\mathcal{C}$ if, for every subset $B \subseteq S$,
there is a set $A \in \mathcal{C}$ such that $S \cap A = B$. The
VC dimension of $\mathcal{C}$ is the largest integer d such
that there is a finite set of cardinality d that is
shattered by $\mathcal{C}$.

Example 3 It can be shown that the set of half-
planes in $\mathbb{R}^2$ has VC dimension two. Choose a
set $S = \{x, y, z\}$ consisting of three points that
are not collinear, as in Fig. 2. Then there are
$2^3 = 8$ subsets of S. The point is to show that
for each of these eight subsets, there is a half-
plane that contains precisely that subset, nothing
more and nothing less. That this is possible is
shown in Fig. 2. Four out of the eight situations
are depicted in this figure, and the remaining four
situations can be covered by taking the comple-
ment of the half-plane shown. It is also necessary
to show that *no set with four or more elements
can be shattered*, but that step is omitted; instead

the reader is referred to any standard text such
as Vidyasagar (1997). More generally, it can be
shown that the set of half-planes in $\mathbb{R}^k$ has VC
dimension $k + 1$.

Example 4 The set of convex polygons has infi-
nite VC dimension. To see this, let S be a strictly
convex set, as shown in Fig. 2b. (Recall that a
set is "strictly convex" if none of its boundary
points is a convex combination of other points in
the set.) Choose any finite collection of boundary
points, call it $S = \{x_1, \ldots, x_n\}$. If B is a
subset of S, then the convex hull of B does not
contain any other point of S, due to the strict
convexity property. Since this argument holds for
every integer n, the class of convex polygons has
infinite VC dimension.

Two Important Theorems

Out of the many important results in learning
theory, two are noteworthy.

Theorem 1 (Blumer et al. (1989)) *A concept
class is distribution-free PAC learnable if and
only if it has finite VC dimension.*

Theorem 2 (Benedek and Itai (1991)) *Suppose
P is a fixed probability distribution. Then the
concept class $\mathcal{C}$ is PAC learnable if and only if,
for every positive number ϵ, it is possible to cover
$\mathcal{C}$ by a finite number of balls of radius ϵ, with
respect to the pseudometric d_P.*

Now let us return to the two examples stud-
ied previously. Since the set of half-planes has
finite VC dimension, it is distribution-free PAC

learnable. The set of convex polygons can be shown to satisfy the conditions of Theorem 2 if P is the uniform distribution and is therefore PAC learnable. However, since it has infinite VC dimension, it follows from Theorem 1 that it is *not* distribution-free PAC learnable.

Summary and Future Directions

This brief entry presents only the most basic aspects of PAC learning theory. Many more results are known about PAC learning theory, and of course many interesting problems remain unsolved. Some of the known extensions are:

- Learning under an "intermediate" family of probability distributions $\mathcal{P}$ that is not necessarily equal to $\mathcal{P}^*$, the set of *all* distributions (Kulkarni and Vidyasagar 1997)
- Relaxing the requirement that the algorithm should work uniformly well for all target concepts and requiring instead only that it should work with high probability (Campi and Vidyasagar 2001)
- Relaxing the requirement that the training samples are independent of each other and permitting them to have Markovian dependence (Gamarnik 2003; Meir 2000) or β-mixing dependence (Vidyasagar 2003)

There is considerable research in finding alternate sets of necessary and sufficient conditions for learnability. Unfortunately, many of these conditions are unverifiable and amount to tautological restatements of the problem under study.

Cross-References

- ▶ Iterative Learning Control
- ▶ Learning in Games
- ▶ Neural Control and Approximate Dynamic Programming
- ▶ Stochastic Games and Learning

Bibliography

Anthony M, Bartlett PL (1999) Neural network learning: theoretical foundations. Cambridge University Press, Cambridge

Anthony M, Biggs N (1992) Computational learning theory. Cambridge University Press, Cambridge

Benedek G, Itai A (1991) Learnability by fixed distributions. Theor Comput Sci 86:377–389

Blumer A, Ehrenfeucht A, Haussler D, Warmuth M (1989) Learnability and the Vapnik-Chervonenkis dimension. J ACM 36(4):929–965

Campi M, Vidyasagar M (2001) Learning with prior information. IEEE Trans Autom Control 46(11):1682–1695

Devroye L, Györfi L, Lugosi G (1996) A probabilistic theory of pattern recognition. Springer, New York

Gamarnik D (2003) Extension of the PAC framework to finite and countable Markov chains. IEEE Trans Inf Theory 49(1):338–345

Kearns M, Vazirani U (1994) Introduction to computational learning theory. MIT, Cambridge

Kulkarni SR, Vidyasagar M (1997) Learning decision rules under a family of probability measures. IEEE Trans Inf Theory 43(1):154–166

Meir R (2000) Nonparametric time series prediction through adaptive model selection. Mach Learn 39(1):5–34

Natarajan BK (1991) Machine learning: a theoretical approach. Morgan-Kaufmann, San Mateo

van der Vaart AW, Wallner JA (1996) Weak convergence and empirical processes. Springer, New York

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Vapnik VN (1998) Statistical learning theory. Wiley, New York

Vidyasagar M (1997) A theory of learning and generalization. Springer, London

Vidyasagar M (2003) Learning and generalization: with applications to neural networks. Springer, London

Lie Algebraic Methods in Nonlinear Control

Matthias Kawski

School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA

Abstract

Lie algebraic methods generalize matrix methods and algebraic rank conditions to smooth nonlinear systems. They capture the essence of noncommuting flows and give rise to noncommutative analogues of Taylor expansions. Lie algebraic

L

rank conditions determine controllability, observability, and optimality. Lie algebraic methods are also employed for state-space realization, control design, and path planning.

Keywords

Baker-Campbell-Hausdorff formula; Chen-Fliess series; Lie bracket

Definition

This article considers generally nonlinear control systems (affine in the control) of the form

$$\begin{aligned} \dot{x} &= f_0(x) + u_1 f_1(x) + \ldots u_m f_m(x) \\ y &= \varphi(x) \end{aligned} \quad (1)$$

where the state x takes values in $\mathbb{R}^n$, or more generally in an n-dimensional manifold M^n, the f_i are smooth vector fields, $\varphi: \mathbb{R}^n \mapsto \mathbb{R}^p$ is a smooth output function, and the controls $u = (u_1, \ldots, u_m): [0, T] \mapsto U$ are piecewise continuous, or, more generally, measurable functions taking values in a closed convex subset $U \subseteq R^m$ that contains 0 in its interior.

Lie algebraic techniques refers to analyzing the system (1) and designing controls and stabilizing feedback laws by employing relations satisfied by iterated Lie brackets of the system vector fields f_i.

Introduction

Systems of the form (1) contain as a special case time-invariant linear systems $\dot{x} = Ax + Bu$, $y = Cx$ (with constant matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$) that are well-studied and are a mainstay of classical control engineering. Properties such as controllability, stabilizability, observability, and optimal control and various others are determined by relationships satisfied by higher-order matrix products of A, B, and C.

Since the early 1970s, it has been well understood that the appropriate generalization of

this matrix algebra, and, e.g., invariant linear subspaces, to nonlinear systems is in terms of the Lie algebra generated by the vector fields f_i, integral submanifolds of this Lie algebra, and the algebra of iterated Lie derivatives of the output function.

The Lie bracket of two smooth vector fields $f, g: M \mapsto TM$ is defined as the vector field $[f, g]: M \mapsto TM$ that maps any smooth function $\varphi \in C^\infty(M)$ to the function $[f, g]\varphi = fg\varphi - gf\varphi$.

In local coordinates, if

$$f(x) = \sum_{i=1}^n f^i(x) \frac{\partial}{\partial x^i} \quad \text{and}$$

$$g(x) = \sum_{i=1}^n g^i(x) \frac{\partial}{\partial x^i},$$

then

$$\begin{aligned} [f, g](x) &= \sum_{i,j=1}^n \left(f^j(x) \frac{\partial g^i}{\partial x^j}(x) \right. \\ &\qquad \left. -g^j(x) \frac{\partial f^i}{\partial x^j}(x) \right) \frac{\partial}{\partial x^i}. \end{aligned}$$

With some abuse of notation, one may abbreviate this to $[f, g] = (Dg)f - (Df)g$, where f and g are considered as column vector fields and Df and Dg denote their Jacobian matrices of partial derivatives.

Note that with this convention the Lie bracket corresponds to the negative of the commutator of matrices: If $P, Q \in \mathbb{R}^{n \times n}$ define, in matrix notation, the linear vector fields $f(x) = Px$ and $g(x) = Qx$, then $[f, g](x) = (QP - PQ)x = -[P, Q]x$.

Noncommuting Flows

Geometrically the Lie bracket of two smooth vector fields f_1 and f_2 is an infinitesimal measure of the lack of commutativity of their flows. For a smooth vector field f and an initial point $x(0) = p \in M$, denote by $e^{tf} p$ the solution of the differential equation $\dot{x} = f(x)$ at time t. Then

$$[f_1, f_2]\varphi(p) = \lim_{t \to 0} \frac{1}{2t^2} \left(\varphi \left(e^{-tf_2} e^{-tf_1} e^{tf_2} e^{tf_1} p \right) - \varphi(p) \right).$$

As a most simple example, consider parallel parking a unicycle, moving it sideways without slipping. Introduce coordinates (x, y, θ) for the location in the plane and the steering angle. The dynamics are governed by $\dot{x} = u_1 \cos \theta$, $\dot{y} = u_1 \sin \theta$, and $\dot{\theta} = u_2$ where the control u_1 is interpreted as the signed rolling speed and u_2 as the angular velocity of the steering angle. Written in the form (1), one has $f_1 = (\cos \theta, \sin \theta, 0)^T$ and $f_2 = (0, 0, 1)^T$. (In this case the drift vector field $f_0 \equiv 0$ vanishes.) If the system starts at $(0, 0, 0)^T$, then via the sequence of control actions of the form *turn left*, *roll forward*, *turn back*, and *roll backwards*, one may steer the system to a point $(0, \Delta y, 0)^T$ with $\Delta y > 0$. This sideways motion corresponds to the value $(0, 1, 0)^T$ of the Lie bracket $[f_1, f_2] = (-\sin \theta, \cos \theta, 0)^T$ at the origin. It encapsulates that steering and rolling do not commute. This example is easily expanded to model, e.g., the sideways motion of a car, or a truck with multiple trailers; see, e.g., Bloch (2003), Bressan and Piccoli (2007), and Bullo and Lewis (2005). In such cases longer iterated Lie brackets correspond to the required more intricate control actions needed to obtain, e.g., a pure sideways motion.

In the case of linear systems, if the Kalman rank condition $\text{rank}[B, AB, A^2B, \ldots A^{n-1}B] = n$ is not satisfied, then all solutions curves of the system starting from the same point $x(0) = p$ are at all times $T > 0$ constrained to lie in a proper affine subspace. In the nonlinear setting the role of the compound matrix of that condition is taken by the Lie algebra $L = L(f_0, f_1, \ldots f_m)$ of all finite linear combinations of iterated Lie brackets of the vector fields f_i. As an immediate consequence of the Frobenius integrability theorem, if at a point $x(0) = p$ the vector fields in L span the whole tangent space, then it is possible to reach an open neighborhood of the initial point by concatenating flows of the system (1) that correspond to piecewise constant controls. Conversely, in the case of analytic vector fields and a compact set U of admissible control values, the Hermann-Nagano theorem guarantees that if the dimension of the subspace $L(p) = \{f(p) : f \in L\} < n$ is not maximal, then all such trajectories are confined to stay in a lower-dimensional proper integral submanifold of L through the point p. For a comprehensive introduction, see, e.g., the textbooks Bressan and Piccoli (2007), Isidori (1995), and Sontag (1998).

Controllability

Define the reachable set $\mathcal{R}_T(p)$ as the set of all terminal points $x(T; u, p)$ at time T of trajectories of (1) that start at the initial point $x(0) = p$ and correspond to admissible controls. Commonly known as the *Lie algebra rank condition* (LARC), the above condition determines whether the system is accessible from the point p, which means that for arbitrarily small time $T > 0$, the reachable set $\mathcal{R}_T(p)$ has nonempty n-dimensional interior. For most applications one desires stronger controllability properties. Most amenable to Lie algebraic methods, and practically relevant, is small-time local controllability (STLC): The system is STLC from p if p lies in the interior of $\mathcal{R}_T(p)$ for every $T > 0$. In the case that there is no drift vector field f_0, accessibility is equivalent to STLC. However, in general, the situation is much more intricate, and a rich literature is devoted to various necessary or sufficient conditions for STLC. A popular such condition is the Hermes condition. For this define the subspaces $\mathcal{S}^1 = \text{span}\{(\text{ad}^j f_0, f_i) : 1 \leq j \leq m, \ j \in \mathbb{Z}^+\}$, and recursively $\mathcal{S}^{k+1} = \text{span}\{[g_1, g_k] : g_1 \in \mathcal{S}^1, \ g_k \in \mathcal{S}^k\}$. Here $(ad^0 f, g) = g$, and recursively $(ad^{k+1} f, g) = [f, (ad^k f, g)]$. The Hermes condition guarantees in the case of analytic vector fields and, e.g., $U = [-1, 1]^m$ that if the system satisfies the (LARC) and for every $k \geq 1$, $\mathcal{S}^{2k}(p) \subseteq \mathcal{S}^{2k-1}(p)$, then the system is (STLC). For more general conditions, see Sussmann (1987) and also Kawski (1990) for a broader discussion.

The importance and value of Lie algebraic conditions may in large part be ascribed to their geometric character, their being invariant under

coordinate changes and feedback. In particular, in the analytic case, the Lie relations completely determine the local properties of the system, in the sense that Lie algebra homomorphism between two systems gives rise to a local diffeomorphism that maps trajectories to trajectories (Sussmann 1974).

Exponential Lie Series

A central analytic tool in Lie algebraic methods that takes the role of Taylor expansions in classical analysis of dynamical system is the Chen-Fliess series which associates to every admissible control $u: [0, T] \mapsto U$ a formal power series

$$\mathrm{CF}(u, T) = \sum_I \int_0^T du^I \cdot X_{i_1} \dots X_{i_s} \quad (2)$$

over a set $\{X_0, X_1, \dots X_m\}$ of noncommuting indeterminates (or *letters*). For every multi-index $I = (i_1, i_2, \dots 1_s) \in \{0, 1, \dots m\}^s$, $s \geq 0$, the coefficient of X_I is the iterated integral defined recursively

$$\int_0^T du^{(I,j)} = \int_0^T \left(\int_0^t u^I \right) du_j(t). \quad (3)$$

Upon evaluating this series via the substitutions $X_i \longleftarrow f_j$, it becomes an *asymptotic series for the propagation of solutions of* (1): For f_j, φ analytic, U compact, p in a compact set, and $T \geq 0$ sufficiently small, one has

$$\varphi(x(t; u, p)) = \sum_I \int_0^T du^I \cdot (f_{i_1} \dots f_{i_s} \varphi)(p). \quad (4)$$

One application of particular interest is to construct approximating systems of a given system (1) that preserve critical geometric properties, but which have an simpler structure. One such class is that of nilpotent systems, that is, systems whose Lie algebra $L = L(f_0, f_1, \dots f_m)$ is nilpotent, and for which solutions can be found by simple quadratures. While truncations of the Chen-Fliess series

never correspond to control systems of the same form, much work has been done in recent years to rewrite this series in more useful formats. For example, the infinite directed exponential product expansion in Sussmann (1986) that uses Hall trees immediately may be interpreted in terms of free nilpotent systems and consequently helps in the construction of nilpotent approximating systems. More recent work, much of it of a combinatorial algebra nature and utilizing the underlying Hopf algebras, further simplifies similar expansions and in particular yields explicit formulas for a continuous Baker-Campbell-Hausdorff formula or for the logarithm of the Chen-Fliess series (Gehrig and Kawski 2008).

Observability and Realization

In the setting of linear systems a well-defined algebraic sense dual to the concept of controllability is that of observability. Roughly speaking the system (1) is observable if knowledge of the output $y(t) = \varphi(x(t; u, p))$ over an arbitrarily small interval suffices to construct the current state $x(t; u, p)$ and indeed the past trajectory $x(\,\cdot\,; u, p)$. In the linear setting observability is equivalent to the rank condition $\mathrm{rank}[C^T, (CA)^T, \dots, (CA^{n-})^T] = n$ being satisfied. In the nonlinear setting, the place of the rows of this compound matrix is taken by the functions in the observation algebra, which consists of all finite linear combinations of iterated Lie derivatives $f_{i_s} \cdots f_{i_1} \varphi$ of the output function.

Similar to the Hankel matrices introduced in the latter setting, in the case of a finite Lie rank, one again can use the output algebra to construct realizations in the form of (1) for systems which are initially only given in terms of input-output descriptions, or in terms of formal Fliess operators; see, e.g., Fliess (1980), Gray and Wang (2002), and Jakubczyk (1986) for further reading.

Optimal Control

In a well-defined geometric way, conditions for optimal control are dual to conditions for controllability and thus are directly amenable to Lie algebraic methods. Instead of considering a separate functional

$$J(u) = \psi(x(T; u, p)) + \int_0^T L(t, x(t; u, p), u(t)) \, dt$$

(5)

to be minimized, it is convenient for our purposes to augment the state by, e.g., defining $\dot{x}_0 = 1$ and $\dot{x}_{n+1} = L(x_0, x, u)$. For example, in the case of time-optimal control, one again obtains an enlarged system of the same form (1); else one utilizes more general Lie algebraic methods that also apply to systems not necessarily affine in the control.

The basic picture for systems with a compact set U of admissible values of the controls involves the attainable funnel $\mathcal{R}_{\leq T}(p)$ consisting of all trajectories of the system (1) starting at $x(0) = p$ that correspond to admissible controls. The trajectory corresponding to an optimal control u^* must at time T lie on the boundary of the funnel $\mathcal{R}_{\leq T}(p)$ and hence also at all prior times (using the invariance of domain property implied by the continuity of the flow). Hence one may associate a covector field along such optimal trajectory that at every time points in the direction of an outward normal. The Pontryagin Maximum Principle is a first-order characterization of such trajectory covector field pairs. Its pointwise maximization condition essentially says that if at any time $t_0 \in [0, T]$ one replaces the optimal control $u^*(\cdot)$ by any admissible control variation on an interval $[t_0, t_0 + \varepsilon]$, then such variation may be transported along the flow to yield, in the limit as $\varepsilon \searrow 0$, an *inward* pointing tangent vector to the reachable set $\mathcal{R}_T(p)$ at $x(T; u^*, p)$. To obtain stronger higher-order conditions for maximality, one may combine several such families of control variations. The effects of such combinations are again calculated in terms of iterated Lie brackets of the vector fields f_i. Indeed, necessary conditions for optimality, for a trajectory to lie on the boundary of the funnel $\mathcal{R}_{\leq T}(p)$, immediately translate into sufficient conditions for STLC, for the initial point to lie in the interior of $\mathcal{R}_T(p)$, and vice versa. For recent work employing Lie algebraic methods in optimality conditions, see, e.g., Agrachev et al. (2002).

Summary and Future Research

Lie algebraic techniques may be seen as a direct generalization of matrix linear algebra tools that have proved so successful in the analysis and design of linear systems. However, in the nonlinear case, the known algebraic rank conditions still exhibit gaps between necessary and sufficient conditions for controllability and optimality. Also, new, not yet fully understood, topological and resonance obstructions stand in the way of controllability implying stabilizability. Systems that exhibit special structure, such as living on Lie groups, or being second order such as typical mechanical systems, are amenable to further refinements of the theory; compare, e.g., the use of affine connections and the symmetric product in Bullo et al. (2000). Other directions of ongoing and future research involve the extension of Lie algebraic methods to infinite dimensional systems and to generalize formulas to systems with less regularity; see, e.g., the work by Rampazzo and Sussmann (2007) on Lipschitz vector fields, thereby establishing closer connections with nonsmooth analysis (Clarke 1983) in control.

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Feedback Linearization of Nonlinear Systems
▶ Lie Algebraic Methods in Nonlinear Control

Bibliography

Agrachev AA, Stefani G, Zezza P (2002) Strong optimality for a bang-bang trajectory. SIAM J Control Optim. 41:991–1014 (electronic)

Bloch AM (2003) Nonholonomic mechanics and control. Interdisciplinary applied mathematics, vol 24. Springer, New York. With the collaboration of J. Baillieul, P. Crouch and J. Marsden, With scientific input from P. S. Krishnaprasad, R. M. Murray and D. Zenkov, Systems and Control.

Bressan A, Piccoli B (2007) Introduction to the mathematical theory of control. AIMS series on applied mathematics, vol 2. American Institute of Mathematical Sciences (AIMS), Springfield

Bullo F, Lewis AD (2005) Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems. Texts in applied mathematics, vol 49. Springer, New York

Bullo F, Leonard NE, Lewis AD (2000) Controllability and motion algorithms for underactuated Lagrangian systems on Lie groups: mechanics and nonlinear control systems. IEEE Trans Autom Control 45:1437–1454

Clarke FH (1983) Optimization and nonsmooth analysis. Canadian mathematical society series of monographs and advanced texts. Wiley, New York. A Wiley-Interscience Publication

Fliess M (1980) Realizations of nonlinear systems and abstract transitive Lie algebras. Bull Am Math Soc (N.S.) 2:444–446

Gehrig E, Kawski M (2008) A hopf-algebraic formula for compositions of noncommuting flows. In: Proceedings IEEE conference decision and cointrol, Cancun, pp 156–1574

Gray WS, Wang Y (2002) Fliess operators on L_p spaces: convergence and continuity. Syst Control Lett 46:67–74

Isidori A (1995) Nonlinear control systems. Communications and control engineering series, 3rd edn. Springer, Berlin

Jakubczyk B (1986) Local realizations of nonlinear causal operators. SIAM J Control Optim 24:230–242

Kawski M (1990) High-order small-time local controllability. In: Nonlinear controllability and optimal control. Monographs and textbooks in pure and applied mathematics, vol 133. Dekker, New York, pp 431–467

Rampazzo F, Sussmann HJ (2007) Commutators of flow maps of nonsmooth vector fields. J Differ Equ 232:134–175

Sontag ED (1998) Mathematical control theory. Texts in applied mathematics, vol 6, 2nd edn. Springer, New York. Deterministic finite-dimensional systems

Sussmann HJ (1974) An extension of a theorem of Nagano on transitive Lie algebras. Proc Am Math Soc 45:349–356

Sussmann HJ (1986) A product expansion for the Chen series. In: Theory and applications of nonlinear control systems (Stockholm, 1985). North-Holland, Amsterdam, pp 323–335

Sussmann HJ (1987) A general theorem on local controllability. SIAM J Control Optim 25:158–194

Linear Matrix Inequality Techniques in Optimal Control

Robert E. Skelton
University of California, San Diego, CA, USA

Abstract

LMI (linear matrix inequality) techniques offer more flexibility in the design of dynamic linear systems than techniques that minimize a scalar functional for optimization. For linear state space models, multiple goals (performance bounds) can be characterized in terms of LMIs, and these can serve as the basis for controller optimization via finite-dimensional convex feasibility problems. LMI formulations of various standard control problems are described in this article, including dynamic feedback stabilization, covariance control, LQR, H_∞ control, L_∞ control, and information architecture design.

Keywords

Control system design; Covariance control; H_∞ control; L_∞ control; LQR/LQG; Matrix inequalities; Sensor/actuator design

Early Optimization History

Hamilton invented state space models of nonlinear dynamic systems with his generalized momenta work in the 1800s (Hamilton 1834, 1835), but at that time the lack of computational tools prevented broad acceptance of the first-order form of dynamic equations. With the rapid development of computers in the 1960s, state space models evoked a formal control theory for minimizing a scalar function of control and state, propelled by the calculus of variations and Pontryagin's maximum principle. Optimal control has been a pillar of control theory for the last 50 years. In fact, all of the problems discussed in this article can perhaps be solved

by minimizing a scalar functional, but a search is required to find the right functional. Globally convergent algorithms are available to do just that for quadratic functionals, but more direct methods are now available.

Since the early 1990s, the focus for linear system design has been to pose control problems as feasibility problems, to satisfy multiple constraints. Since then, feasibility approaches have dominated design decisions, and such feasibility problems may be convex or not. If the problem can be reduced to a set of linear matrix inequalities (LMIs) to solve, then convexity is proven. However, failure to find such LMI formulations of the problem does not mean it is not convex, and computer-assisted methods for convex problems are available to avoid the search for LMIs (see Camino et al. 2003).

In the case of linear dynamic models of stochastic processes, optimization methods led to the popularization of linear quadratic Gaussian (LQG) optimal control, which had globally optimal solutions (see Skelton 1988). The first two moments of the stochastic process (the mean and the covariance) can be controlled with these methods, even if the distribution of the random variables involved is not Gaussian. Hence, LQG became just an acronym for the solution of quadratic functionals of control and state variables, even when the stochastic processes were not Gaussian. The label LQG was often used even for deterministic problems, where a time integral, rather than an expectation operator, was minimized, with given initial conditions or impulse excitations. These were formally called LQR (linear quadratic regulator) problems. Later the book (Skelton 1988) gave the formal conditions under which the LQG and the LQR answers were numerically identical, and this particular version of LQR was called the *deterministic LQG*.

It was always recognized that the quadratic form of the state and control in the LQG problem was an artificial goal. The real control goals usually involved prespecified performance bounds on *each* of the outputs and bounds on *each* channel of control. This leads to matrix inequalities (MIs) rather than scalar minimizations. While

it was known early that *any* stabilizing linear controller could be obtained by some choice of weights in an LQG optimization problem (see Chap. 6 and references in Skelton 1988), it was not known until the 1980s *what* particular choice of weights in LQG would yield a solution to the matrix inequality (MI) problem. See early attempts in Skelton (1988), and see Zhu and Skelton (1992) and Zhu et al. (1997) for a globally convergent algorithm to find such LQG weights when the MI problem has a solution. Since then, rather than stating a minimization problem for a meaningless sum of outputs and inputs, linear control problems can now be stated simply in terms of norm bounds on *each* input vector and/or *each* output vector of the system (L_2 bounds, L_∞ bounds, or variance bounds and covariance bounds). These *feasibility* problems are convex for state feedback or full-order output feedback controllers (the focus of this elementary introduction), and these can be solved using linear matrix inequalities (LMIs), as illustrated in this article. However, the earliest approach to these MI problems was iterative LQG solutions (to find the correct weights to use in the quadratic penalty of the state), as in Skelton (1988), Zhu and Skelton (1992), and Zhu et al. (1997).

Matrix Inequalities

Let $\mathbf{Q}$ be any square matrix. The linear matrix inequality (LMI) "$\mathbf{Q} > \mathbf{0}$" is just a short-hand notation to represent a certain scalar inequality. That is, the matrix notation "$\mathbf{Q} > \mathbf{0}$" means "the scalar $\mathbf{x^T Q x}$ is positive for all values of $\mathbf{x}$, except $\mathbf{x} = \mathbf{0}$." Obviously this is a property of $\mathbf{Q}$, not $\mathbf{x}$, hence the abbreviated matrix notation $\mathbf{Q} > \mathbf{0}$. This is called a linear matrix inequality (LMI), since the matrix unknown $\mathbf{Q}$ appears linearly in the inequality $\mathbf{Q} > \mathbf{0}$. Note also that any square matrix $\mathbf{Q}$ can be written as the sum of a symmetric matrix $\mathbf{Q_s} = \frac{1}{2}(\mathbf{Q} + \mathbf{Q^T})$, and a skew-symmetric matrix $\mathbf{Q_k} = \frac{1}{2}(\mathbf{Q} - \mathbf{Q^T})$, but $\mathbf{x^T Q_k x} = \mathbf{0}$, so only the symmetric part of the matrix $\mathbf{Q}$ affects the scalar $\mathbf{x^T Q x}$. We assume hereafter without loss of generality that $\mathbf{Q}$ is

L

symmetric. The notation "$\mathbf{Q} \geq \mathbf{0}$" means "the scalar $\mathbf{x^TQx}$ cannot be negative for any $\mathbf{x}$."

Lyapunov proved that $\mathbf{x(t)}$ converges to zero if there exists a matrix $\mathbf{Q}$ such that, along the nonzero trajectory of a dynamic system (e.g., the system $\dot{\mathbf{x}} = \mathbf{Ax}$), two scalars have the property, $\mathbf{x(t)^TQx(t)} > 0$ and $d/dt(\mathbf{x^T(t)Qx(t)}) < 0$. This proves that the following statements are all equivalent:

1. For any initial condition $\mathbf{x(0)}$ of the system $\dot{\mathbf{x}} = \mathbf{Ax}$, the state $\mathbf{x(t)}$ converges to zero.
2. All eigenvalues of $\mathbf{A}$ lie in the open left half plane.
3. There exists a matrix $\mathbf{Q}$ with the two properties $\mathbf{Q} > \mathbf{0}$ and $\mathbf{QA + A^TQ < 0}$.
4. The set of all quadratic Lyapunov functions that can be used to prove the stability or instability of the null solution of $\dot{\mathbf{x}} = \mathbf{Ax}$ is given by $\mathbf{x^TQ^{-1}x}$, where $\mathbf{Q}$ is any square matrix with the two properties of item 3 above.

LMIs are prevalent throughout the fundamental concepts of control theory, such as controllability and observability. For the linear system example $\dot{\mathbf{x}} = \mathbf{Ax + Bu}$, $\mathbf{y} = \mathbf{Cx}$, the "Observability Gramian" is the infinite integral $\mathbf{Q} = \int \mathbf{e^{A^Tt}C^TCe^{At}dt}$. Furthermore $\mathbf{Q} > \mathbf{0}$ if and only if $(\mathbf{A, C})$ is an observable pair, and $\mathbf{Q}$ is bounded only if the observable modes are asymptotically stable. When it exists, the solution of $\mathbf{QA + A^TQ + C^TC = 0}$ satisfies $\mathbf{Q} > \mathbf{0}$ if and only if the matrix pair $(\mathbf{A, C})$ is observable.

Likewise the "Controllability Gramian" $\mathbf{X} = \int \mathbf{e^{At}BB^Te^{A^Tt}dt} > \mathbf{0}$ if and only if the pair $(\mathbf{A, B})$ is controllable. If $\mathbf{X}$ exists, it satisfies $\mathbf{XA^T + AX + BB^T = 0}$, and $\mathbf{X} > \mathbf{0}$ if and only if $(\mathbf{A, B})$ is a controllable pair. Note also that the matrix pair $(\mathbf{A, B})$ is controllable for any $\mathbf{A}$ if $\mathbf{BB^T > 0}$, and the matrix pair $(\mathbf{A, C})$ is observable for any $\mathbf{A}$ if $\mathbf{C^TC > 0}$. Hence, the existence of $\mathbf{Q} > \mathbf{0}$ or $\mathbf{X} > \mathbf{0}$ satisfying either $(\mathbf{QA + A^TQ < 0})$ or $(\mathbf{AX + XA^T < 0})$ is equivalent to the statement that "all eigenvalues of $\mathbf{A}$ lie in the open left half plane."

It should now be clear that the set of all stabilizing state feedback controllers, $\mathbf{u = Gx}$, is parametrized by the inequalities $\mathbf{Q} > \mathbf{0}$, $\mathbf{Q(A + BG) + (A + BG)^TQ < 0}$. The difficulty in this

MI is the appearance of the product of the two unknowns $\mathbf{Q}$ and $\mathbf{G}$, so more work is required to show how to use LMIs to solve this problem.

In the sequel some techniques are borrowed from linear algebra, where a linear matrix equality (LME) $\mathbf{\Gamma G\Lambda = \Theta}$ may or may not have a solution $\mathbf{G}$. For LMEs there are two separate questions to answer. The first question is "Does there exist a solution?" and the answer is "if and only if $\mathbf{\Gamma\Gamma^+\Theta\Lambda^+\Lambda = \Theta}$." The second question is "What is the set of all solutions?" and the answer is "$\mathbf{G = \Gamma^+\Theta\Lambda^+ + Z - \Gamma^+\Gamma Z\Lambda\Lambda^+}$, where $\mathbf{Z}$ is arbitrary, and the $+$ symbol denotes Pseudo Inverse." LMI approaches employ the same two questions by formulating the necessary and sufficient conditions for the existence of an LMI solution and then to parametrize all solutions.

Perhaps the earliest book on LMI control methods was Boyd et al. (1994), but the results and notations used herein are taken from Skelton et al. (1998). Other important LMI papers and books can give the reader a broader background, including Iwasaki and Skelton (1994), Gahinet and Apkarian (1994), de Oliveira et al. (2002), Li et al. (2008), de Oliveira and Skelton (2001), Camino et al. (2001, 2003), Boyd and Vandenberghe (2004), Iwasaki et al. (2000), Khargonekar and Rotea (1991), Vandenberghe and Boyd (1996), Scherer (1995), Scherer et al. (1997), Balakrishnan et al. (1994), Gahinet et al. (1995), and Dullerud and Paganini (2000).

Control Design Using LMIs

Consider the feedback control system

$$\begin{bmatrix} \mathbf{\dot{x}_p} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{A_p} & \mathbf{D_p} & \mathbf{B_p} \\ \mathbf{C_p} & \mathbf{D_y} & \mathbf{B_y} \\ \mathbf{M_p} & \mathbf{D_z} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x_p} \\ \mathbf{w} \\ \mathbf{u} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{\dot{x}_c} \end{bmatrix} = \begin{bmatrix} \mathbf{D_c} & \mathbf{C_c} \\ \mathbf{B_c} & \mathbf{A_c} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{x_c} \end{bmatrix} = \mathbf{G} \begin{bmatrix} \mathbf{z} \\ \mathbf{x_c} \end{bmatrix}, \quad (1)$$

where $\mathbf{z}$ is the measurement vector, $\mathbf{y}$ is the output to be controlled, $\mathbf{u}$ is the control vector, $\mathbf{x_p}$ is the plant state vector, $\mathbf{x_c}$ is the state of the controller,

and **w** is the external disturbance (in some cases below we treat **w** as a zero-mean white noise). We seek to choose the control matrix **G** to satisfy the given upper bounds on the output covariance $E[\mathbf{yy^T}] \leq \bar{\mathbf{Y}}$, where E represents the steady-state expectation operator in the stochastic case (i.e., when **w** is white noise), and in the deterministic case E represents the infinite integral of the matrix $[\mathbf{yy^T}]$. The math is the same in each case, with appropriate interpretations of certain matrices. For a rigorous equivalence of the deterministic and stochastic interpretations, see Skelton (1988). By defining the matrices,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_p} \\ \mathbf{x_c} \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{A_{cl}} & \mathbf{B_{cl}} \\ \mathbf{C_{cl}} & \mathbf{D_{cl}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{D} \\ \mathbf{C} & \mathbf{F} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{H} \end{bmatrix} \mathbf{G} \begin{bmatrix} \mathbf{M} & \mathbf{E} \end{bmatrix} \quad (2)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_p} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B_p} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M_p} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D_p} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{D_z} \\ \mathbf{0} \end{bmatrix}$$
(3)

$$\mathbf{C} = \begin{bmatrix} \mathbf{C_p} & \mathbf{0} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{B_y} & \mathbf{0} \end{bmatrix}, \quad \mathbf{F} = \mathbf{D_y}, \quad (4)$$

one can write the closed-loop system dynamics in the form

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{cl}} & \mathbf{B_{cl}} \\ \mathbf{C_{cl}} & \mathbf{D_{cl}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}. \quad (5)$$

Often it is of interest to characterize the set of all controllers that can satisfy performance bounds on both the outputs and inputs, $E[\mathbf{yy^T}] \leq \bar{\mathbf{Y}}$ and $E[\mathbf{uu^T}] \leq \bar{\mathbf{U}}$, and we call these *covariance* control problems. But without prespecified performance bounds $\bar{\mathbf{Y}}, \bar{\mathbf{U}}$, one can require stability only. Such examples are given below.

Many Control Problems Reduce to the Same LMI

Let the left (right) null spaces of any matrix **B** be defined by matrices $\mathbf{U_B}$ ($\mathbf{V_B}$), where $\mathbf{U_B^T B} = \mathbf{0}$, $\mathbf{U_B^T U_B} > \mathbf{0}$, ($\mathbf{BV_B} = \mathbf{0}$, $\mathbf{V_B^T V_B} > \mathbf{0}$). For

any given matrices $\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{\Theta}$, Chap. 9 of the book (Skelton et al. 1998) provides all **G** which solve

$$\mathbf{\Gamma G \Lambda} + (\mathbf{\Gamma G \Lambda})^\mathbf{T} + \mathbf{\Theta} < \mathbf{0}, \quad (6)$$

and proves that there exists such a matrix **G** if and only if the following two conditions hold:

$$\mathbf{U_\Gamma^T \Theta} U_\mathbf{\Gamma} < \mathbf{0}, \quad \text{or} \quad \mathbf{\Gamma \Gamma^T} > \mathbf{0}, \quad (7)$$

$$\mathbf{V_\Lambda^T \Theta} V_\mathbf{\Lambda} < \mathbf{0}, \quad \text{or} \quad \mathbf{\Lambda^T \Lambda} > \mathbf{0}. \quad (8)$$

If **G** exists, then one set of such **G** is given by

$$\mathbf{G} = -\rho\mathbf{\Gamma^T \Phi \Lambda^T} (\mathbf{\Lambda \Phi \Lambda^T})^{-1}, \ \mathbf{\Phi} = (\rho\mathbf{\Gamma \Gamma^T} - \mathbf{\Theta})^{-1},$$
(9)

where $\rho > 0$ is an arbitrary scalar such that

$$\mathbf{\Phi} = (\rho\mathbf{\Gamma \Gamma^T} - \mathbf{\Theta})^{-1} > \mathbf{0}. \quad (10)$$

All **G** which solve the problem are given by Theorem 2.3.12 in Skelton et al. (1998). As elaborated in Chap. 9 of Skelton et al. (1998), 17 different control problems (using either state feedback or full-order dynamic controllers) all reduce to this same mathematical problem. That is, by defining the appropriate $\mathbf{\Theta}, \mathbf{\Lambda}, \mathbf{\Gamma}$, a very large number of different control problems, including the characterization of all stabilizing controllers, covariance control, H-infinity control, L-infinity control, LQG control, and H_2 control, can be reduced to the *same* matrix inequality (13). Several examples from Skelton et al. (1998) follow.

Stabilizing Control
There exists a controller **G** that stabilizes the system (1) if and only if (7) and (8) hold, where the matrices are defined by

$$\begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Lambda^T} & \mathbf{\Theta} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{XM^T} & \mathbf{AX} + \mathbf{XA^T} \end{bmatrix}.$$
(11)

One can also write such results in another way, as in Corollary 6.2.1 of Skelton et al. (1998, p. 135): There exists a control of the form $\mathbf{u} = \mathbf{Gx}$ that can stabilize the system $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$ if and only if there exists a matrix $\mathbf{X} > \mathbf{0}$

satisfying $\mathbf{B}^{\perp}(\mathbf{AX} + \mathbf{XA^T})(\mathbf{B}^{\perp})^{\mathbf{T}} < \mathbf{0}$, where $\mathbf{B}^{\perp}$ denotes the left null space of $\mathbf{B}$. In this case all stabilizing controllers may be parametrized by $\mathbf{G} = -\mathbf{B^T P} + \mathbf{L Q^{1/2}}$, for any $\mathbf{Q} > \mathbf{0}$ and a $\mathbf{P} > \mathbf{0}$ satisfying $\mathbf{PA} + \mathbf{A^T P} - \mathbf{PBB^T P} + \mathbf{Q} = \mathbf{0}$. The matrix $\mathbf{L}$ is any matrix that satisfies the norm bound $\|\mathbf{L}\| < 1$. Youla et al. (1976) provided a parametrization of the set of all stabilizing controllers, but the parametrization was infinite dimensional (as it did not impose any restriction on the order or form of the controller). So for finite calculations one had to truncate the set to a finite number before optimization or stabilization started. As noted above, on the other hand, all stabilizing state feedback controllers $\mathbf{G}$ can be parametrized in terms of an arbitrary but finite-dimensional norm-bounded matrix $\mathbf{L}$. Similar results apply for the dynamic controllers of any fixed order (see Chap. 6 in Skelton et al. 1998).

Covariance Upper Bound Control

In the system (1), suppose that $\mathbf{D_y} = \mathbf{0}$, $\mathbf{B_y} = \mathbf{0}$ and that $\mathbf{w}$ is zero-mean white noise with intensity $\mathbf{I}$. Let a required upper bound $\overline{\mathbf{Y}} > \mathbf{0}$ on the steady-state output covariance $\mathbf{Y} = E[\mathbf{yy^T}]$ be given. The following statements are equivalent:

(i) There exists a controller $\mathbf{G}$ that solves the covariance upper bound control problem $\mathbf{Y} < \overline{\mathbf{Y}}$.

(ii) There exists a matrix $\mathbf{X} > \mathbf{0}$ such that $\mathbf{Y} = \mathbf{CXC^T} < \overline{\mathbf{Y}}$ and (7) and (8) hold, where the matrices are defined by

$$\begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^{\mathbf{T}} & \boldsymbol{\Theta} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{B} & \mathbf{XM^T} & \mathbf{AX} + \mathbf{XA^T} & \mathbf{D} \\ \mathbf{0} & \mathbf{E^T} & \mathbf{D^T} & -\mathbf{I} \end{bmatrix} \tag{12}$$

($\boldsymbol{\Theta}$ occupies the last two columns).

Proof is provided by Theorem 9.1.2 in Skelton et al. (1998).

Linear Quadratic Regulator

Consider the linear time-invariant system (1). Suppose that $\mathbf{D_y} = \mathbf{0}$, $\mathbf{D_z} = \mathbf{0}$ and that $\mathbf{w}$ is the impulsive disturbance $\mathbf{w(t)} = \mathbf{w_0}\delta(\mathbf{t})$. Let a performance bound $\gamma > 0$ be given, where

the required performance is to keep the integral squared output ($\|\mathbf{y}\|_{\mathbf{L2}}^{\mathbf{2}}$) less than the prespecified value $\|\mathbf{y}\|_{\mathbf{L2}} < \gamma$ for any vector $\mathbf{w_0}$ such that $\mathbf{w_0^T w_0} \leq \mathbf{1}$, and $\mathbf{x_0} = \mathbf{0}$. This problem is labeled linear quadratic regulator (LQR). The following statements are equivalent:

(i) There exists a controller $\mathbf{G}$ that solves the LQR problem.

(ii) There exists a matrix $\mathbf{Y} > \mathbf{0}$ such that $\|\mathbf{D^T YD}\| < \gamma^2$ and (7) and (8) hold, where the matrices are defined by

$$\begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^{\mathbf{T}} & \boldsymbol{\Theta} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{YB} & \mathbf{M^T} & \mathbf{YA} + \mathbf{A^T Y} & \mathbf{M^T} \\ \mathbf{H} & \mathbf{0} & \mathbf{M} & -\mathbf{I} \end{bmatrix}. \tag{13}$$

Proof is provided by Theorem 9.1.3 in Skelton et al. (1998).

H_∞ Control

LMI techniques provided the first papers to solve the general H_∞ problem, without any restrictions on the plant. See Iwasaki and Skelton (1994) and Gahinet and Apkarian (1994).

Let the closed-loop transfer matrix from $\mathbf{w}$ to $\mathbf{y}$ with the controller in (1) be denoted by $\mathbf{T(s)}$:

$$\mathbf{T(s)} = \mathbf{C_{cl}(sI} - \mathbf{A_{cl})^{-1} B_{cl}} + \mathbf{D_{cl}}. \tag{14}$$

The H_∞ control problem can be defined as follows:

> Let a performance bound $\gamma > 0$ be given. Determine whether or not there exists a controller $\mathbf{G}$ in (1) which asymptotically stabilizes the system and yields the closed-loop transfer matrix (14) such that the peak value of the frequency response is less than γ. That is, $\|\mathbf{T}\|_{\mathbf{H_\infty}} = \sup \|\mathbf{T}(j\omega)\| < \gamma$.

For the H_∞ control problem, we have the following result. Let a required H_∞ performance bound $\gamma > 0$ be given. The following statements are equivalent:

(i) A controller $\mathbf{G}$ solves the H_∞ control problem.

(ii) There exists a matrix $\mathbf{X} > \mathbf{0}$ such that (7) and (8) holds, where

$$\begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^{\mathbf{T}} & \boldsymbol{\Theta} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{B} & \mathbf{XM}^{\mathbf{T}} & \mathbf{AX} + \mathbf{XA}^{\mathbf{T}} & \mathbf{XC}^{\mathbf{T}} & \mathbf{D} \\ \mathbf{H} & \mathbf{0} & \mathbf{CX} & -\gamma\mathbf{I} & \mathbf{F} \\ \mathbf{0} & \mathbf{E}^{\mathbf{T}} & \mathbf{D}^{\mathbf{T}} & \mathbf{F}^{\mathbf{T}} & -\gamma\mathbf{I} \end{bmatrix}$$
(15)

($\boldsymbol{\Theta}$ occupies the last three columns).
Proof is provided by Theorem 9.1.5 in Skelton et al. (1998).

L_∞ Control

The peak value of the frequency response is controlled by the above H_∞ controller. A similar theorem can be written to control the peak in the time domain.

Define $\sup \mathbf{y(t)}^{\mathbf{T}}\mathbf{y(t)} = \|\mathbf{y}\|_{\mathbf{L}\infty}^2$, and let the statement $\|\mathbf{y}\|_{L\infty} < \gamma$ mean that the peak value of $\mathbf{y(t)}^{\mathbf{T}}\mathbf{y(t)}$ is less than γ^2. Suppose that $\mathbf{D_y} = \mathbf{0}$ and $\mathbf{B_y} = \mathbf{0}$. There exists a controller $\mathbf{G}$ which maintains $\|\mathbf{y}\|_{L\infty} < \gamma$ in the presence of any energy-bounded input $\mathbf{w(t)}$ (i.e., $\int_0^\infty \mathbf{w^Tw}dt \leq \mathbf{1}$) if and only if there exists a matrix $\mathbf{X} > \mathbf{0}$ such that $\mathbf{CXC^T} < \gamma^2\mathbf{I}$ and (7) and (8) hold, where

$$\begin{bmatrix} \boldsymbol{\Gamma} & \boldsymbol{\Lambda}^{\mathbf{T}} & \boldsymbol{\Theta} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{B} & \mathbf{XM}^{\mathbf{T}} & \mathbf{AX} + \mathbf{XA}^{\mathbf{T}} & \mathbf{D} \\ \mathbf{0} & \mathbf{E}^{\mathbf{T}} & \mathbf{D}^{\mathbf{T}} & -\gamma\mathbf{I} \end{bmatrix}.$$
(16)

Proof is provided by Theorem 9.1.4 in Skelton et al. (1998).

Information Architecture in Estimation and Control Problems

In the typical "control problem" that occupies most research literature, the sensors and actuators have already been selected. Yet the selection of sensors and actuators and their locations greatly affect the ability of the control system to do its job efficiently. Perhaps in one location a high-precision sensor is needed, and in another location high precision is not needed, and paying for high precision in that location would therefore be a waste of resources. These decisions must be influenced by the control dynamics which are yet

to be designed. How does one know where to effectively spend money to improve the system? To answer this question, we must optimize the information architecture jointly with the control law.

Let us consider the problem of selecting the control law jointly with the selection of the precision (defined here as the inverse of the noise intensity) of each actuator/sensor, subject to the constraint of specified upper bounds on the covariance of output error and control signals, and specified upper bounds on the sensor/actuator cost. We assume the cost of these devices is proportional to their precision (i.e., the cost is equal to the *price per unit of precision*, times the precision). Traditionally, with full-order controllers, and *prespecified* sensor/actuator instruments (with specified precisions); this is a well-known solved convex problem (which means it can be converted to an LMI problem if desired), see Chap. 6 of Skelton et al. (1998). If we enlarge the domain of the freedom to include sensor/actuator precisions, it is not obvious whether the feasibility problem is convex or not. The following shows that this problem of including the sensor/actuator precisions within the control design problem is indeed convex and therefore completely solved. The proof is provided in Li et al. (2008).

Consider the linear control system (1)–(5). Assume that the cost of sensors and actuators is proportional to their precision, which we herein define to be the inverse of the noise intensity (or variance, in the discrete-time case). So if the price per unit of precision of the i-th sensor/actuator is P_{ii}, and if the variance (or intensity) of the noise associated with the i-th sensor/actuator is W_{ii}, then the total cost of all sensors and actuators is $\sum P_{ii}W_{ii}^{-1}$, or simply $\mathbf{tr}(\mathbf{PW^{-1}})$, where $\mathbf{P} = \mathrm{diag}(P_{ii})$ and $\mathbf{W^{-1}} = \mathrm{diag}(W_{ii}^{-1})$.

Consider the control system (1). Suppose that $\mathbf{D_y} = \mathbf{0}$, $\mathbf{B_y} = \mathbf{0}$, $\mathbf{w} = \begin{bmatrix} \mathbf{w_s^T} & \mathbf{w_a^T} \end{bmatrix}^{\mathbf{T}}$ is the zero-mean sensor/actuator noise, $\mathbf{D_p} = [\mathbf{0} \quad \mathbf{D_a}]$ and $\mathbf{D_z} = [\mathbf{D_s} \quad \mathbf{0}]$. If the $\bar{\$}$ represents the allowed upper bound on sensor/actuator costs, there exists a dynamic controller $\mathbf{G}$ that satisfies the constraints

$$E[\mathbf{uu^T}] < \bar{\mathbf{U}}, \quad E[\mathbf{yy^T}] < \bar{\mathbf{Y}}, \quad \mathbf{tr}(\mathbf{PW^{-1}}) < \bar{\$}$$
(17)

in the presence of sensor/actuator noise with intensity $\text{diag}(W_{ii}) = \mathbf{W}$ (which like $\mathbf{G}$ should be considered a design variable not fixed a priori) if and only if there exist matrices $\mathbf{L}, \mathbf{F}, \mathbf{Q}, \mathbf{X}, \mathbf{Z}, \mathbf{W^{-1}}$ such that

$$\mathbf{tr}(\mathbf{PW^{-1}}) < \bar{\$} \qquad (18)$$

$$\begin{bmatrix} \bar{\mathbf{Y}} & \mathbf{C_pX} & \mathbf{C_p} \\ (\mathbf{C_pX})^\mathbf{T} & \mathbf{X} & \mathbf{I} \\ \mathbf{C_p}^\mathbf{T} & \mathbf{I} & \mathbf{Z} \end{bmatrix} > \mathbf{0}, \quad \begin{bmatrix} \bar{\mathbf{U}} & \mathbf{L} & \mathbf{0} \\ \mathbf{L^T} & \mathbf{X} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{Z} \end{bmatrix} > \mathbf{0},$$

$$\begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{21}^\mathbf{T} \\ \mathbf{\Phi}_{21} & -\mathbf{W^{-1}} \end{bmatrix} < 0, \qquad (19)$$

$$\mathbf{\Phi}_{21} = \begin{bmatrix} \mathbf{D_a} & \mathbf{0} \\ \mathbf{ZD_a} & \mathbf{FD_s} \end{bmatrix},$$

$$\phi = \begin{bmatrix} \mathbf{A_pX} + \mathbf{B_pL} & \mathbf{A_p} \\ \mathbf{Q} & \mathbf{ZA_p} + \mathbf{FM_p} \end{bmatrix},$$

$$\mathbf{\Phi}_{11} = \phi + \phi^\mathbf{T}. \qquad (20)$$

Note that the matrix inequalities (18)–(20) are LMIs in the collection of variables $(\mathbf{L}, \mathbf{F}, \mathbf{Q}, \mathbf{X}, \mathbf{Z}, \mathbf{W^{-1}})$, whereby joint control/sensor/actuator design is a convex problem.

Assume a solution $(\mathbf{L}, \mathbf{F}, \mathbf{Q}, \mathbf{X}, \mathbf{Z}, \mathbf{W})$ is found for the LMIs (18)–(20). Then the problem (17) is solved by the controller

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{V_l^{-1}} & -\mathbf{V_l^{-1}ZB_p} \end{bmatrix} \begin{bmatrix} \mathbf{Q} - \mathbf{ZA_pX} & \mathbf{F} \\ \mathbf{L} & \mathbf{0} \end{bmatrix} \\ \begin{bmatrix} \mathbf{0} & \mathbf{V_r^{-1}} \\ \mathbf{I} & -\mathbf{M_pXV_r^{-1}} \end{bmatrix}, \qquad (21)$$

where $\mathbf{V_l}$ and $\mathbf{V_r}$ are left and right factors of the matrix $\mathbf{I} - \mathbf{YX}$ (which can be found from the singular value decomposition $\mathbf{I} - \mathbf{YX} = \mathbf{U\Sigma V^T} = (\mathbf{U\Sigma^{1/2}})(\mathbf{\Sigma^{1/2}V^T}) = (\mathbf{V_l})(\mathbf{V_r})$).

To emphasize the theme of this article, to relate optimization to LMIs, we note that three optimization problems present themselves in the above problem with three constraints: control effort $\bar{\mathbf{U}}$, output performance $\bar{\mathbf{Y}}$, and instrument costs $\bar{\$}$. To solve optimization problems, one can fix any two of these prespecified upper bounds and iteratively reduce the level set value of the third "constraint" until feasibility is lost. This process minimizes the resource expressed by the third constraint, while enforcing the other two constraints.

As an example, if cost is not a concern, one can always set large limits for $\bar{\$}$ and discover the best assignment of sensor/actuator precisions for the specified performance requirements. These precisions produced by the algorithm are the values W_{ii}^{-1}, produced from the solution (18)–(20), where the observed rankings $W_{ii}^{-1} > W_{jj}^{-1} > W_{kk}^{-1} > \ldots$ indicate which sensors or actuators are most critical to the required performance goals $(\bar{\mathbf{U}}, \bar{\mathbf{Y}}, \bar{\$})$. If any precision W_{nn}^{-1} is essentially zero, compared to other required precisions, then the math is asserting that the information from this sensor (n) is not important for the control objectives specified, or the control signals through this actuator channel (n) are ineffective in controlling the system to these specifications. This information leads us to a technique for choosing the best sensor actuators and their location.

The previous discussion provides the precisions (W_{ii}^{-1}) required of each sensor and each actuator in the system. Our final application of this theory locates sensors and actuators in a large-scale system, by discarding the least effective ones. Suppose we solve any of the above feasibility problems, by starting with the entire admissible set of sensors and actuators (without regard to cost). For example, in a flexible structure control problem we might not know whether to place a rate sensor or displacement sensors at a given location, so we add both. We might not know whether to use torque or force actuators, so we add both. We fill up the system with all the possibilities we might want to consider, and let the above precision rankings (available after the above LMI problem is solved) reveal how much precision is needed at each location and at each sensor/actuator. If there is a large gap in the precisions required (say $W_{11}^{-1} > W_{22}^{-1} > W_{33}^{-1} >> \ldots W_{nn}^{-1}$), then delete the sensor/actuator n and repeat the LMI problem with one less sensor or actuator. Continue deleting sensors/actuators in

this manner until feasibility of the problem is lost. Then this algorithm, stopping at the previous iteration, has selected the best distribution of sensors/actuators for solving the specific problem specified by the allowable bounds $(\bar{\$}, \bar{U}, \bar{Y})$. The most important contribution of the above algorithm has been to extend control theory to solve system design problems that involve more than just deigning control gains. This enlarges the set of solved linear control problems, from solutions of linear controllers with sensors/actuators prespecified to solutions which specify the sensor/actuator requirements jointly with the control solution.

Summary

LMI techniques provide more powerful tools for designing dynamic linear systems than techniques that minimize a scalar functional for optimization, since multiple goals (bounds) can be achieved for *each* of the outputs and inputs. Optimal control has been a pillar of control theory for the last 50 years. In fact, all of the problems discussed in this article can perhaps be solved by minimizing a scalar functional, but a search is required to find the right functional. Globally convergent algorithms are available to do just that for quadratic functionals. But more direct methods are now available (since the early 1990s) for satisfying multiple constraints. Since then, feasibility approaches have dominated design decisions (at least for linear systems), and such feasibility problems may be convex or not. If the problem can be reduced to a set of LMIs to solve, then convexity is proven. However, failure to find such LMI formulations of the problem does not mean it is not convex, and computer-assisted methods for convex problems are available to avoid the search for LMIs (see Camino et al. 2003). Optimization can also be achieved with LMI methods by reducing the level set for one of the bounds, while maintaining all the other bounds. This level set is reduced iteratively, between convex (LMI) solutions, until feasibility is lost. A most amazing fact is that most of the common linear control design

problems all reduce to exactly the same matrix inequality problem (6). The set of such equivalent problems includes LQR, the set of all stabilizing controllers, the set of all H_∞ controllers, and the set of all L_∞ controllers. The discrete and robust versions of these problems are also included in this equivalent set; 17 control problems have been found to be equivalent to LMI problems.

LMI techniques extend the range of solvable system design problems beyond just control design. By integrating information architecture and control design, one can simultaneously choose the control gains and the precision required of all sensor/actuators to satisfy the closed-loop performance constraints. These techniques can be used to select the information (with precision requirements) required to solve a control or estimation problem, using the best economic solution (minimal precision). For a more complete discussion of LMI problems in control, read Dullerud and Paganini (2000), de Oliveira et al. (2002), Li et al. (2008), de Oliveira and Skelton (2001), Gahinet and Apkarian (1994), Iwasaki and Skelton (1994), Camino et al. (2001, 2003), Skelton et al. (1998), Boyd and Vandenberghe (2004), Boyd et al. (1994), Iwasaki et al. (2000), Khargonekar and Rotea (1991), Vandenberghe and Boyd (1996), Scherer (1995), Scherer et al. (1997), Balakrishnan et al. (1994), and Gahinet et al. (1995).

Cross-References

▶ H-Infinity Control
▶ H_2 Optimal Control
▶ Linear Quadratic Optimal Control
▶ LMI Approach to Robust Control
▶ Stochastic Linear-Quadratic Control

Bibliography

Balakrishnan V, Huang Y, Packard A, Doyle JC (1994) Linear matrix inequalities in analysis with multipliers. In: Proceedings of the 1994 American control conference, Baltimore, vol 2, pp 1228–1232

Boyd SP, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Boyd SP, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia

Camino JF, Helton JW, Skelton RE, Ye J (2001) Analysing matrix inequalities systematically: how to get schur complements out of your life. In: Proceedings of the 5th SIAM conference on control & its applications, San Diego

Camino JF, Helton JW, Skelton RE, Ye J (2003) Matrix inequalities: a symbolic procedure to determine convexity automatically. Integral Equ Oper Theory 46(4):399–454

de Oliveira MC, Skelton RE (2001) Stability tests for constrained linear systems. In: Reza Moheimani SO (ed) Perspectives in robust control. Lecture notes in control and information sciences. Springer, New York, pp 241–257. ISBN:1852334525

de Oliveira MC, Geromel JC, Bernussou J (2002) Extended H_2 and H_∞ norm characterizations and controller parametrizations for discrete-time systems. Int J Control 75(9):666–679

Dullerud G, Paganini F (2000) A course in robust control theory: a convex approach. Texts in applied mathematics. Springer, New York

Gahinet P, Apkarian P (1994) A linear matrix inequality approach to H_∞ control. Int J Robust Nonlinear Control 4(4):421–448

Gahinet P, Nemirovskii A, Laub AJ, Chilali M (1995) LMI control toolbox user's guide. The Mathworks Inc., Natick

Hamilton WR (1834) On a general method in dynamics; by which the study of the motions of all free systems of attracting or repelling points is reduced to the search and differentiation of one central relation, or characteristic function. Philos Trans R Soc (part II):247–308

Hamilton WR (1835) Second essay on a general method in dynamics. Philos Trans R Soc (part I):95–144

Iwasaki T, Skelton RE (1994) All controllers for the general H_∞ control problem – LMI existence conditions and state-space formulas. Automatica 30(8):1307–1317

Iwasaki T, Meinsma G, Fu M (2000) Generalized S-procedure and finite frequency KYP lemma. Math Probl Eng 6:305–320

Khargonekar PP, Rotea MA (1991) Mixed H_2/H_∞ control: a convex optimization approach. IEEE Trans Autom Control 39:824–837

Li F, de Oliveira MC, Skelton RE (2008) Integrating information architecture and control or estimation design. SICE J Control Meas Syst Integr 1(2):120–128

Scherer CW (1995) Mixed H_2/H_∞ control. In: Isidori A (ed) Trends in control: a European perspective, pp 173–216. Springer, Berlin

Scherer CW, Gahinet P, Chilali M (1997) Multiobjective output-feedback control via LMI optimization. IEEE Trans Autom Control 42(7):896–911

Skelton RE (1988) Dynamics Systems Control: linear systems analysis and synthesis. Wiley, New York

Skelton RE, Iwasaki T, Grigoriadis K (1998) A unified algebraic approach to control design. Taylor & Francis, London

Vandenberghe L, Boyd SP (1996) Semidefinite programming. SIAM Rev 38:49–95

Youla DC, Bongiorno JJ, Jabr HA (1976) Modern Wiener-Hopf design of optimal controllers, part ii: the multivariable case. IEEE Trans Autom Control 21:319–338

Zhu G, Skelton R (1992) A two-Riccati, feasible algorithm for guaranteeing output l_∞ constraints. J Dyn Syst Meas Control 114(3):329–338

Zhu G, Rotea M, Skelton R (1997) A convergent algorithm for the output covariance constraint control problem. SIAM J Control Optim 35(1):341–361

Linear Quadratic Optimal Control

H.L. Trentelman
Johann Bernoulli Institute for Mathematics and
Computer Science, University of Groningen,
Groningen, AV, The Netherlands

Abstract

Linear quadratic optimal control is a collective term for a class of optimal control problems involving a linear input-state-output system and a cost functional that is a quadratic form of the state and the input. The aim is to minimize this cost functional over a given class of input functions. The optimal input depends on the initial condition, but can be implemented by means of a state feedback control law independent of the initial condition. Both the feedback gain and the optimal cost can be computed in terms of solutions of Riccati equations.

Keywords

Introduction

Linear quadratic optimal control is a generic term that collects a number of optimal control problems for linear input-state-output systems in which a quadratic cost functional is minimized over a given class of input functions. This functional is formed by integrating a quadratic form of the state and the input over a finite or an infinite time interval. Minimizing the energy of the output over a finite or infinite time interval can be formulated in this framework and in fact provides a major motivation for this class of optimal control problems. A common feature of the solutions to the several versions of the problem is that the optimal input functions can be given in the form of a linear state feedback control law. This makes it possible to implement the optimal controllers as a feedback loop around the system. Another common feature is that the optimal value of the cost functional is a quadratic form of the initial condition on the system. This quadratic form is obtained by taking the appropriate solution of a Riccati differential equation or algebraic Riccati equation associated with the system.

Systems with Inputs and Outputs

Consider the continuous-time, linear time-invariant input-output system in state space form represented by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad z(t) = Cx(t) + Du(t). \tag{1}$$

This system will be referred to as Σ. In (1), A, B, C, and D are maps between suitable spaces (or matrices of suitable dimensions) and the functions x, u, and z are considered to be defined on the real axis $\mathbb{R}$ or on any subinterval of it. In particular, one often assumes the domain of definition to be the nonnegative part of $\mathbb{R}$. The function u is called the *input*, and its values are assumed to be given from outside the system. The class of admissible input functions will be denoted **U**. Often, **U** will be the class of piecewise

continuous or locally integrable functions, but for most purposes, the exact class from which the input functions are chosen is not important. We will assume that input functions take values in an m-dimensional space $\mathcal{U}$, which we often identify with $\mathbb{R}^m$. The variable x is called the *state variable* and it is assumed to take values in an n-dimensional space $\mathcal{X}$. The space $\mathcal{X}$ will be called the *state space*. It will usually be identified with $\mathbb{R}^n$. Finally, z is called the *to be controlled output* of the system and takes values in a p-dimensional space $\mathcal{Z}$, which we identify with $\mathbb{R}^p$. The solution of the differential equation of Σ with initial value $x(0) = x_0$ will be denoted as $x_u(t, x_0)$. It can be given explicitly using the variation-of-constants formula (see Trentelman et al. 2001, p. 38). The set of eigenvalues of a given matrix M is called the *spectrum* of M and is denoted by $\sigma(M)$. The system (1) is called *stabilizable* if there exists a map (matrix of suitable dimensions) F such that $\sigma(A + BF) \subset \mathbb{C}^-$. Here, $\mathbb{C}^-$ denotes the open left half complex plane, i.e., $\{\lambda \in \mathbb{C} \mid \text{Re}(\lambda) < 0\}$. We often express this property by saying that *the pair (A, B) is stabilizable*.

The Linear Quadratic Optimal Control Problem

Assume that our aim is to keep all components of the output $z(t)$ as small as possible, for all $t \geq 0$. In the ideal situation, with initial state $x(0) = 0$, the uncontrolled system (with control input $u = 0$) evolves along the stationary solution $x(t) = 0$. Of course, the output $z(t)$ will then also be equal to zero for all t. If, however, at time $t = 0$ the state of the system is perturbed to, say, $x(0) = x_0$, with $x_0 \neq 0$, then the uncontrolled system will evolve along a state trajectory unequal to the stationary zero solution, and we will get $z(t) = Ce^{At}x_0$. To remedy this, from time $t = 0$ on, we can apply an input function u, so that for $t \geq 0$ the corresponding output becomes equal to $z(t) = Cx_u(t, x_0) + Du(t)$. Keeping in mind that we want the output $z(t)$ to be as small as possible for all $t \geq 0$, we can measure its size by the quadratic cost functional

$$J(x_0, u) = \int_0^\infty \|z(t)\|^2 \, dt, \qquad (2)$$

where $\| \cdot \|$ denotes the Euclidean norm. Our aim to keep the values of the output as small as possible can then be expressed as requiring this integral to be as small as possible by suitable choice of input function u. In this way we arrive at the *linear quadratic optimal control problem*:

Problem 1 Consider the system $\Sigma : \dot{x}(t) = Ax(t) + Bu(t)$, $z(t) = Cx(t) + Du(t)$. Determine for every initial state x_0 an input $u \in \mathbf{U}$ (a space of functions $[0, \infty) \to \mathcal{U}$) such that

$$J(x_0, u) := \int_0^\infty \|z(t)\|^2 \, dt \qquad (3)$$

is minimal. Here $z(t)$ denotes the output trajectory $z_u(t, x_0)$ of Σ corresponding to the initial state x_0 and input function u.

Since the system is linear and the integrand in the cost functional is a quadratic function of z, the problem is called *linear quadratic*. Of course, $\|z\|^2 = x^\top C^\top C x + 2u^\top D^\top C x + u^\top D^\top D u$, so the integrand can also be considered as a quadratic function of (x, u). The convergence of the integral in (3) is of course a point of concern. Therefore, one often considers the corresponding *finite-horizon* problem in a preliminary investigation. In this problem, a final time T is given and one wants to minimize the integral

$$J(x_0, u, T) := \int_0^T \|z(t)\|^2 \, dt. \qquad (4)$$

In contrast to this, the first problem above is sometimes called the *infinite horizon problem*. An important issue is also the *convergence of the state*. Obviously, convergence of the integral does not always imply the convergence to zero of the state. Therefore, distinction is made between the problem with zero and with free endpoint. Problem 1 as stated is referred to as the *problem with free endpoint*. If one restricts the inputs u in the problem to those for which the resulting state trajectory tends to zero, one speaks about the *problem with zero endpoint*. Specifically:

Problem 2 In the situation of Problem 1, determine for every initial state x_0 an input $u \in \mathbf{U}$ such that $x_u(t, x_0) \to 0$ $(t \to \infty)$ and such that under this condition, $J(x_0, u)$ is minimized.

In the literature various special cases of these problems have been considered, and names have been associated to these special cases. In particular, Problems 1 and 2 are called *regular* if the matrix D is injective, equivalently, $D^\top D > 0$. If, in addition, $C^\top D = 0$ and $D^\top D = I$, then the problems are said to be in *standard form*. In the standard case, the integrand in the cost functional reduces to $\|z\|^2 = x^\top C^\top C x + u^\top u$. We often write $Q = C^\top C$. The standard case is a special case, which is not essentially simpler than the general regular problem, but which gives rise to simpler formulas. The general regular problem can be reduced to the standard case by means of a suitable feedback transformation.

The Finite-Horizon Problem

The finite-horizon problem in standard from is formulated as follows:

Problem 3 Given the system $\dot{x}(t) = Ax(t) + Bu(t)$, a final time $T > 0$, and symmetric matrices N and Q such that $N \geq 0$ and $Q \geq 0$, determine for every initial state x_0 a piecewise continuous input function $u : [0, T] \to \mathcal{U}$ such that the integral

$$J(x_0, u, T) := \int_0^T x(t)^\top Q x(t) + u(t)^\top u(t) \, dt$$
$$+ x(T)^\top N x(T) \qquad (5)$$

is minimized.

In this problem, we have introduced a *weight on the final state*, using the matrix N. This generalization of the problem does not give rise to additional complications.

A key ingredient in solving this finite-horizon problem is the *Riccati differential equation* associated with the problem:

$$\dot{P}(t) = A^\top P(t) + P(t)A - P(t)BB^\top P(t) + Q,$$
$$P(0) = N. \qquad (6)$$

This is a quadratic differential equation on the interval $[0, \infty]$ in terms of the matrices A, B, and Q, and with initial condition given by the weight matrix N on the final state. The unknown in the differential equation is the matrix valued function $P(t)$. The following theorem solves the finite-horizon problem. It states that the Riccati differential equation with initial condition (6) has a unique solution on $[0, \infty)$, that the optimal value of the cost functional is determined by the value of this solution at time T, and that there exists a unique optimal input that is generated by a time-varying state feedback control law:

Theorem 1 *Consider Problem 3. The following properties hold:*

1. *The Riccati differential equation with initial value* (6) *has a unique solution $P(t)$ on $[0, \infty)$. This solution is symmetric and positive semidefinite for all $t \geq 0$.*

2. *For each x_0 there is exactly one optimal input function, i.e., a piecewise continuous function u^* on $[0, T]$ such that $J(x_0, u^*, T) = J^*(x_0, T) := \inf\{J(x_0, u, T) \mid u \in \mathbf{U}\}$. This optimal input function u^* is generated by the time-varying feedback control law*

$$u(t) = -B^\top P(T - t)x(t) \quad (0 \leq t \leq T).$$
(7)

3. *For each x_0, the minimal value of the cost functional equals*

$$J^*(x_0, T) = x_0^\top P(T)x_0.$$

4. *If $N = 0$, then the function $t \mapsto P(t)$ is an increasing function in the sense that $P(t) - P(s)$ is positive semidefinite for $t \geq s$.*

The Infinite-Horizon Problem with Free Endpoint

We consider the situation as described in Theorem 1 with $N = 0$. An obvious conjecture is that $x_0^\top P(T)x_0$ converges to the minimal cost of the infinite-horizon problem as $T \to \infty$. The convergence of $x_0^\top P(T)x_0$ for all x_0 is equivalent to the convergence of the matrix $P(T)$ for

$T \to \infty$ to some matrix P^-. Such a convergence does not always take place. In order to achieve convergence, we make the following assumption: for every x_0, there exists an input u for which the integral

$$J(x_0, u) := \int_0^\infty x(t)^\top Q x(t) + u(t)^\top u(t) \, dt$$
(8)

converges, i.e., for which the cost $J(x_0, u)$ is finite. Obviously, for the problem to make sense for all x_0, this condition is necessary. It is easily seen that the stabilizability of (A, B) is a sufficient condition for the above assumption to hold (not necessary, take, e.g., $Q = 0$). Take an arbitrary initial state x_0 and assume that $\bar{u}$ is a function such that the integral (8) is finite. We have for every $T > 0$ that

$$x_0^\top P(T)x_0 \leq J(x_0, \bar{u}, T) \leq J(x_0, \bar{u}),$$

which implies that for every x_0, the expression $x_0^\top P(T)x_0$ is bounded. This implies that $P(T)$ is bounded. Since $P(T)$ is increasing with respect to T, it follows that $P^- := \lim_{T \to \infty} P(T)$ exists. Since P satisfies the differential equation (6), it follows that also $\dot{P}(t)$ has a limit as $t \to \infty$. It is easily seen that this latter limit must be zero. Hence, $P = P^-$ satisfies the following equation:

$$A^\top P + PA - PBB^\top P + Q = 0. \quad (9)$$

This is called the *algebraic Riccati equation* (ARE). The solutions of this equation are exactly the constant solutions of the Riccati differential equation. The previous consideration shows that the ARE has a positive semidefinite solution P^-. The solution is not necessarily unique, not even with the extra condition that $P \geq 0$. However, P^- turns out to be the *smallest* real symmetric positive semidefinite solution of the ARE.

The following theorem now establishes a complete solution to the regular standard form version of Problem 1:

L

Theorem 2 *Consider the system* $\dot{x}(t) = Ax(t) + Bu(t)$ *together with the cost functional*

$$J(x_0, u) := \int_0^\infty x(t)^\top Q x(t) + u(t)^\top u(t)\, dt,$$

with $Q \geq 0$. *Factorize* $Q = C^\top C$. *Then, the following statements are equivalent:*

1. *For every* $x_0 \in \mathcal{X}$, *there exists* $u \in \mathbf{U}$ *such that* $J(x_0, u) < \infty$.
2. *The ARE* (9) *has a real symmetric positive semidefinite solution* P.

Assume that one of the above conditions holds. Then, there exists a smallest real symmetric positive semidefinite solution of the ARE, i.e., there exists a real symmetric solution $P^- \geq 0$ *such that for every real symmetric solution* $P \geq 0$, *we have* $P^- \leq P$. *For every* x_0, *we have*

$$J^*(x_0) := \inf\{J(x_0, u) \mid u \in \mathbf{U}\} = x_0^\top P^- x_0.$$

Furthermore, for every x_0, *there is exactly one optimal input function, i.e., a function* $u^* \in \mathbf{U}$ *such that* $J(x_0, u^*) = J^*(x_0)$. *This optimal input is generated by the time-invariant feedback law*

$$u(t) = -B^\top P^- x(t).$$

The Infinite-Horizon Problem with Zero Endpoint

In addition to the free endpoint problem, we consider the version of the linear quadratic problem with zero endpoint. In this case the aim is to minimize for every x_0 the cost functional over all inputs u such that $x_u(t, x_0) \to 0$ $(t \to \infty)$. For each x_0 such u exists if and only if the pair (A, B) is stabilizable. A solution to the regular standard form version of Problem 2 is stated next:

Theorem 3 *Consider the system* $\dot{x}(t) = Ax(t) + Bu(t)$ *together with the cost functional*

$$J(x_0, u) := \int_0^\infty x(t)^\top Q x(t) + u(t)^\top u(t)\, dt,$$

with $Q \geq 0$. *Assume that* (A, B) *is stabilizable. Then:*

1. *There exists a largest real symmetric solution of the ARE, i.e., there exists a real symmetric solution* P^+ *such that for every real symmetric solution* P, *we have* $P \leq P^+$. P^+ *is positive semidefinite.*
2. *For every initial state* x_0, *we have*

$$J_0^*(x_0) = x_0^\top P^+ x_0.$$

3. *For every initial state* x_0, *there exists an optimal input function, i.e., a function* $u^* \in \mathbf{U}$ *with* $x(\infty) = 0$ *such that* $J(x_0, u^*) = J_0^*(x_0)$ *if and only if every eigenvalue of* A *on the imaginary axis is* (Q, A) *observable, i.e.,*
   $$\mathrm{rank} \begin{pmatrix} A - \lambda I \\ Q \end{pmatrix} = n \text{ for all } \lambda \in \sigma(A) \text{ with}$$
   $\mathrm{Re}(\lambda) = 0$.

   Under this assumption we have:
4. *For every initial state* x_0, *there is exactly one optimal input function* u^*. *This optimal input function is generated by the time-invariant feedback law*

$$u(t) = -B^\top P^+ x(t).$$

5. *The optimal closed-loop system* $\dot{x}(t) = (A - BB^\top P^+)x(t)$ *is stable. In fact,* P^+ *is the unique real symmetric solution of the ARE for which* $\sigma(A - BB^\top P^+) \subset \mathbb{C}^-$.

Summary and Future Directions

Linear quadratic optimal control deals with finding an input function that minimizes a quadratic cost functional for a given linear system. The cost functional is the integral of a quadratic form in the input and state variable of the system. If the integral is taken over, a finite time interval the problem is called a finite-horizon problem, and the optimal cost and optimal state feedback gain can be expressed in terms of the solution of an associated Riccati differential equation. If we integrate over an infinite time interval, the problem is called an infinite-horizon problem. The optimal cost and optimal feedback gain for the free endpoint problem can be found in terms of

the smallest nonnegative real symmetric solution of the associated algebraic Riccati equation. For the zero endpoint problem, these are given in terms of the largest real symmetric solution of the algebraic Riccati equation.

Cross-References

▸ Generalized Finite-Horizon Linear-Quadratic Optimal Control
▸ H-Infinity Control
▸ H$_2$ Optimal Control
▸ Linear State Feedback

Recommended Reading

The linear quadratic regulator problem and the Riccati equation were introduced by R.E. Kalman in the early 1960s (see Kalman 1960). Extensive treatments of the problem can be found in the textbooks Brockett (1969), Kwakernaak and Sivan (1972), and Anderson and Moore (1971). For a detailed study of the Riccati differential equation and the algebraic Riccati equation, we refer to Wonham (1968). Extensions of the linear quadratic regulator problem to linear quadratic optimization problems, where the integrand of the cost functional is a possibly indefinite quadratic function of the state and input variable, were studied in the classical paper of Willems (1971). A further reference for the geometric classification of all real symmetric solutions of the algebraic Riccati equation is Coppel (1974). For the question what level of system performance can be obtained if, in the cost functional, the weighting matrix of the control input is singular or nearly singular leading to singular and nearly singular linear quadratic optimal control problems and "cheap control" problems, we refer to Kwakernaak and Sivan (1972). An early reference for a discussion on the singular problem is the work of Clements and Anderson (1978). More details can be found in Willems (1971) and Schumacher (1983). In singular problems, in general one allows for distributions as inputs. This approach was

worked out in detail in Hautus and Silverman (1983) and Willems et al. (1986). For a more recent reference, including an extensive list of references, we refer to the textbook of Trentelman et al. (2001).

Bibliography

Anderson BDO, Moore JB (1971) Linear optimal control. Prentice Hall, Englewood Cliffs

Brockett RW (1969) Finite dimensional linear systems. Wiley, New York

Clements DJ, Anderson BDO (1978) Singular optimal control: the linear quadratic problem. Volume 5 of lecture notes in control and information sciences. Springer, New York

Coppel WA (1974) Matrix quadratic equations. Bull Aust Math Soc 10:377–401

Hautus MLJ, Silverman LM (1983) System structure and singular control. Linear Algebra Appl 50:369–402

Kalman RE (1960) Contributions to the theory of optimal control. Bol Soc Mat Mex 5:102–119

Kwakernaak H, Sivan R (1972) Linear optimal control theory. Wiley, New York

Schumacher JM (1983) The role of the dissipation matrix in singular optimal control. Syst Control Lett 2:262–266

Trentelman HL, Hautus MLJ, Stoorvogel AA (2001) Control theory for linear systems. Springer, London

Willems JC (1971) Least squares stationary optimal control and the algebraic Riccati equation. IEEE Trans Autom Control 16:621–634

Willems JC, Kitapçi A, Silverman LM (1986) Singular optimal control: a geometric approach. SIAM J Control Optim 24:323–337

Wonham WM (1968) On a matrix Riccati equation of stochastic control. SIAM J Control Optim 6(4): 681–697

L

Linear Quadratic Zero-Sum Two-Person Differential Games

Pierre Bernhard
INRIA-Sophia Antipolis-Méditerranée, Sophia Antipolis, France

Abstract

As in optimal control theory, linear quadratic (LQ) differential games (DG) can be solved, even in high dimension, via a Riccati equation.

However, contrary to the control case, existence of the solution of the Riccati equation is not necessary for the existence of a closed-loop saddle point. One may "survive" a particular, nongeneric, type of conjugate point. An important application of LQDGs is the so-called H_∞-optimal control, appearing in the theory of robust control.

Keywords

Differential games; Finite horizon; H-infinity control; Infinite horizon

Perfect State Measurement

Linear quadratic differential games are a special case of differential games (DG). See the article ▶ Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games. They were first investigated by Ho et al. (1965), in the context of a linearized pursuit-evasion game. This subsection is based upon Bernhard (1979, 1980). A linear quadratic DG is defined as

$$\dot{x} = Ax + Bu + Dv, \quad x(t_0) = x_0,$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $v \in \mathbb{R}^\ell$, $u(\cdot) \in L^2([0, T], \mathbb{R}^m)$, $v(\cdot) \in L^2([0, T], \mathbb{R}^\ell)$. Final time T is given, there is no terminal constraint, and using the notation $x^t K x = \|x\|_K^2$,

$$J(t_0, x_0; u(\cdot), v(\cdot)) = \|x(T)\|_K^2 + \int_{t_0}^T (x^t \; u^t \; v^t)$$

$$\begin{pmatrix} Q & S_1 & S_2 \\ S_1^t & R & 0 \\ S_2^t & 0 & -\Gamma \end{pmatrix} \begin{pmatrix} x \\ u \\ v \end{pmatrix} dt.$$

The matrices of appropriate dimensions, A, B, D, Q, S_i, R, and Γ, may all be measurable functions of time. R and Γ must be positive definite with inverses bounded away from zero. To get the most complete results available, we assume also that K and Q are nonnegative definite, although this is only necessary for some of the following results. Detailed results without that assumption were obtained by Zhang (2005) and Delfour (2005). We chose to set the cross term in uv in the criterion null; this is to simplify the results and is not necessary. This problem satisfies Isaacs' condition (see article DG) even with nonzero such cross terms.

Using the change of control variables

$$u = \tilde{u} - R^{-1} S_1^t x, \qquad v = \tilde{v} + \Gamma^{-1} S_2^t x,$$

yields a DG with the same structure, with modified matrices A and Q, but without the cross terms in xu and xv. (This extends to the case with nonzero cross terms in uv.) Thus, without loss of generality, we will proceed with $(S_1 \; S_2) = (0 \; 0)$.

The existence of open-loop and closed-loop solutions to that game is ruled by two Riccati equations for symmetric matrices P and $P^\star$, respectively, and by a pair of canonical equations that we shall see later:

$$\dot{P} + PA + A^t P - PBR^{-1} B^t P + PD\Gamma^{-1} D^t P$$
$$+ Q = 0, P(T) = K, \tag{1}$$

$$\dot{P}^\star + P^\star A + A^t P^\star + P^\star D\Gamma^{-1} D^t P^\star + Q = 0,$$
$$P^\star(T) = K. \tag{2}$$

When both Riccati equations have a solution over $[t, T]$, it holds that in the partial ordering of definiteness,

$$0 \le P(t) \le P^\star(t).$$

When the saddle point exists, it is represented by the state feedback strategies

$$u = \varphi^\star(t, x) = -R^{-1} B^t P(t)x,$$

$$v = \psi^\star(t, x) = \Gamma^{-1} D^t P(t)x. \tag{3}$$

The control functions generated by this pair of feedbacks will be noted $\hat{u}(\cdot)$ and $\tilde{v}(\cdot)$.

Theorem 1
- *A sufficient condition for the existence of a closed-loop saddle point, then given by*

$(\varphi^\star, \psi^\star)$ in (*3*), is that Eq. (*1*) has a solution $P(t)$ defined over $[t_0, T]$.

- *A necessary and sufficient condition for the existence of an open-loop saddle point is that Eq. (*2*) has a solution over $[t_0, T]$ (and then so does (*1*)). In that case, the pairs $(\hat{u}(\cdot), \hat{v}(\cdot))$, $(\hat{u}(\cdot), \psi^\star)$, and $(\varphi^\star, \psi^\star)$ are saddle points.*
- *A necessary and sufficient condition for $(\varphi^\star, \hat{v}(\cdot))$ to be a saddle point is that Eq. (*1*) has a solution over $[t_0, T]$.*
- *In all cases where a saddle point exists, the Value function is $V(t, x) = \|x\|_{P(t)}^2$.*

However, Eq. (*1*) may fail to have a solution and a closed-loop saddle point still exists. The precise necessary condition is as follows: let $X(\cdot)$ and $Y(\cdot)$ be two square matrix function solutions of the *canonical equations*

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \end{pmatrix} = \begin{pmatrix} A & -BR^{-1}B^t + D\Gamma^{-1}D^t \\ -Q & -A^t \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix},$$
$$\begin{pmatrix} X(T) \\ Y(T) \end{pmatrix} = \begin{pmatrix} I \\ K \end{pmatrix}.$$

The matrix $P(t)$ exists for $t \in [t_0, T]$ if and only if $X(t)$ is invertible over that range, and then, $P(t) = Y(t)X^{-1}(t)$. Assume that the rank of $X(t)$ is piecewise constant, and let $X^\dagger(t)$ denote the pseudo-inverse of $X(t)$ and $\mathcal{R}(X(t))$ its range.

Theorem 2 *A necessary and sufficient condition for a closed-loop saddle point to exist, which is then given by (*3*) with $P(t) = Y(t)X^\dagger(t)$, is that*
1. *$x_0 \in \mathcal{R}(X(t_0))$.*
2. *For almost all $t \in [t_0, T]$, $\mathcal{R}(D(t)) \subset \mathcal{R}(X(t))$.*
3. *$\forall t \in [t_0, T]$, $Y(t)X^\dagger(t) \geq 0$.*

In a case where $X(t)$ is only singular at an isolated instant $t^\star$ (then conditions 1 and 2 above are automatically satisfied), called a *conjugate point* but where YX^{-1} remains positive definite on both sides of it, the conjugate point is called *even*. The feedback gain $F = -R^{-1}B^t P$ diverges upon reaching $t^\star$, but on a trajectory generated by this feedback, the control $u(t) =$

$F(t)x(t)$ remains finite. (See an example in Bernhard 1979.)

If $T = \infty$, with all system and payoff matrices constant and $Q > 0$, Mageirou (1976) has shown that if the algebraic Riccati equation obtained by setting $\dot{P} = 0$ in (*1*) admits a positive definite solution P, the game has a Value $\|x\|_P^2$, but (*3*) may not be a saddle point. ($\psi^\star$ may not be an equilibrium strategy.)

H_∞-Optimal Control

This subsection is entirely based upon Başar and Bernhard (1995). It deals with imperfect state measurement, using Bernhard's nonlinear *minimax certainty equivalence principle* (Bernhard and Rapaport 1996).

Several problems of robust control may be brought to the following one: a linear, time-invariant system with two inputs (control input $u \in \mathbb{R}^m$ and disturbance input $w \in \mathbb{R}^\ell$) and two outputs (measured output $y \in \mathbb{R}^p$ and controlled output $z \in \mathbb{R}^q$) is given. One wishes to control the system with a nonanticipative controller $u(\cdot) = \phi(y(\cdot))$ in order to minimize the induced linear operator norm between spaces of square-integrable functions, of the resulting operator $w(\cdot) \mapsto z(\cdot)$.

It turns out that the problem which has a tractable solution is a kind of dual one: given a positive number γ, is it possible to make this norm no larger than γ? The answer to this question is yes if and only if

$$\inf_{\phi \in \Phi} \sup_{w(\cdot) \in L^2} \int_{-\infty}^{\infty} (\|z(t)\|^2 - \gamma^2 \|w(t)\|^2)\, dt \leq 0.$$

We shall extend somewhat this classical problem by allowing either a time variable system, with a finite horizon T, or a time-invariant system with an infinite horizon.

The dynamical system is

$$\dot{x} = Ax + Bu + Dw, \tag{4}$$

$$y = Cx + \qquad Ew, \tag{5}$$

$$z = Hx + Gu. \tag{6}$$

We let

$$\begin{pmatrix} D \\ E \end{pmatrix} (D^t \ E^t) = \begin{pmatrix} M & L \\ L^t & N \end{pmatrix} ,$$

$$\begin{pmatrix} H^t \\ G^t \end{pmatrix} (H \ G) = \begin{pmatrix} Q & S \\ S^t & R \end{pmatrix} ,$$

and we assume that E is onto, $\Leftrightarrow N > 0$, and G is one-to-one $\Leftrightarrow R > 0$.

Finite Horizon

In this part, we consider a time-varying system, with all matrix functions measurable. Since the state is not known exactly, we assume that the initial state is not known either. The issue is therefore to decide whether the criterion

$$J_\gamma = \|x(T)\|_K^2 + \int_{t_0}^{T} (\|z(t)\|^2 - \gamma^2 \|w(t)\|^2)$$

$$\mathrm{d}t - \gamma^2 \|x_0\|_{\Sigma_0}^2 \qquad (7)$$

may be kept finite and with which strategy. Let

$$\gamma^\star = \inf\{\gamma \mid \inf_{\phi \in \Phi} \sup_{x_0 \in \mathbb{R}^n, w(\cdot) \in L^2} J_\gamma < \infty\} .$$

Theorem 3 $\gamma \leq \gamma^\star$ *if and only if the following three conditions are satisfied:*

1. *The following Riccati equation has a solution over* $[t_0, T]$:

$$-\dot{P} = PA + A^t P - (PB + S)R^{-1}(B^t P + S^t)$$

$$+ \gamma^{-2} PMP + Q , \quad P(T) = K . \quad (8)$$

2. *The following Riccati equation has a solution over* $[t_0, T]$:

$$\dot{\Sigma} = A\Sigma + \Sigma A^t - (\Sigma C^t + L)N^{-1}(C\Sigma + L^t)$$

$$+ \gamma^{-2}\Sigma Q \Sigma + M , \quad \Sigma(t_0) = \Sigma_0 . \quad (9)$$

3. *The following spectral radius condition is satisfied:*

$$\forall t \in [t_0, T], \quad \rho(\Sigma(t)P(t)) < \gamma^2 . \quad (10)$$

In that case, the optimal controller ensuring $\inf_\phi \sup_{x_0, w} J_\gamma$ *is given by a "worst case state"* $\hat{x}(\cdot)$ *satisfying* $\hat{x}(0) = 0$ *and*

$$\dot{\hat{x}} = [A - BR^{-1}(B^t P + S^t) + \gamma^{-2}D(D^t P + L^t)]$$

$$\hat{x} + (I - \gamma^{-2}\Sigma P)^{-1}(\Sigma C^t + L)(y - C\hat{x}) , \quad (11)$$

and the certainty equivalent *controller*

$$\phi^\star(y(\cdot))(t) = -R^{-1}(B^t P + S^t)\hat{x}(t) . \quad (12)$$

Infinite Horizon

The infinite horizon case is the traditional H_∞-optimal control problem reformulated in a state space setting. We let all matrices defining the system be constant. We take the integral in (7) from $-\infty$ to $+\infty$, with no initial or terminal term of course. We add the hypothesis that the pairs (A, B) and (A, D) are stabilizable and the pairs (C, A) and (H, A) detectable. Then, the theorem is as follows:

Theorem 4 $\gamma \leq \gamma^\star$ *if and only if the following conditions are satisfied: The algebraic Riccati equations obtained by placing $\dot{P} = 0$ and $\dot{\Sigma} = 0$ in (8) and (9) have positive definite solutions, which satisfy the spectral radius condition (10). The optimal controller is given by Eqs. (11) and (12), where P and Σ are the minimal positive definite solutions of the algebraic Riccati equations, which can be obtained as the limit of the solutions of the differential equations as $t \rightarrow -\infty$ for P and $t \rightarrow \infty$ for Σ.*

Conclusion

The similarity of the H_∞-optimal control theory with the LQG, stochastic, theory is in many respects striking, as is the duality observation control. Yet, the "observer" of H_∞-optimal control does *not* arise from some estimation theory but from the analysis of a "worst case." The best explanation might be in the duality of the ordinary, or $(+, \times)$, algebra with the idempotent $(\max, +)$ algebra (see Bernhard 1996).

The complete theory of H_∞-optimal control in that perspective has yet to be written.

Cross-References

Bibliography

Başar T, Bernhard P (1995) H^∞-optimal control and related minimax design problems: a differential games approach, 2nd edn. Birkhäuser, Boston

Bernhard P (1979) Linear quadratic zero-sum two-person differential games: necessary and sufficient condition. JOTA 27(1):51–69

Bernhard P (1980) Linear quadratic zero-sum two-person differential games: necessary and sufficient condition: comment. JOTA 31(2):283–284

Bernhard P (1996) A separation theorem for expected value and feared value discrete time control. COCV 1:191–206

Bernhard P, Rapaport A (1996) Min-max certainty equivalence principle and differential games. Int J Robust Nonlinear Control 6:825–842

Delfour M (2005) Linear quadratic differential games: saddle point and Riccati equation. SIAM J Control Optim 46:750–774

Ho Y-C, Bryson AE, Baron S (1965) Differential games and optimal pursuit-evasion strategies. IEEE Trans Autom Control AC-10:385–389

Mageirou EF (1976) Values and strategies for infinite time linear quadratic games. IEEE Trans Autom Control AC-21:547–550

Zhang P (2005) Some results on zero-sum two-person linear quadratic differential games. SIAM J Control Optim 43:2147–2165

Linear State Feedback

Panos J. Antsaklis[1] and A. Astolfi[2,3]
[1]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
[2]Department of Electrical and Electronic Engineering, Imperial College London, London, UK
[3]Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

Abstract

Feedback is a fundamental mechanism in nature and central in the control of systems. The state contains important system information, and applying a control law that uses state information is a very powerful control policy. To illustrate the effect of feedback in linear systems, continuous-time and discrete-time state variable descriptions are used: these allow one to write explicitly the resulting closed-loop descriptions and to study the effect of feedback on the eigenvalues of the closed-loop system. The eigenvalue assignment problem is also discussed.

Keywords

Introduction

Feedback is a fundamental mechanism arising in nature. Feedback is also common in engineered systems and is essential in the automatic control of dynamic processes with uncertainties in their model descriptions and in their interactions with the environment. When feedback is used, the actual values of the system variables are sensed, fed back, and used to control the system. That is, a control law decision process is based not only

on predictions on the system behavior derived from a process model (as in open-loop control) but also on information about the actual behavior (closed-loop feedback control).

Linear Continuous-Time Systems

Consider, to begin with, time-invariant systems described by the state variable description

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du, \quad (1)$$

in which $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the input, $y(t) \in \mathbb{R}^p$ is the output, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are constant matrices. In this case, the linear state feedback (lsf) control law is selected as

$$u(t) = Fx(t) + r(t), \qquad (2)$$

where $F \in \mathbb{R}^{m \times n}$ is the constant gain matrix and $r(t) \in \mathbb{R}^m$ is a new external input.

Substituting (2) into (1) yields the closed-loop state variable description, namely,

$$\begin{aligned}\dot{x} &= (A + BF)x + Br, \\ y &= (C + DF)x + Dr.\end{aligned} \qquad (3)$$

Appropriately selecting F, primarily to modify $A + BF$, one affects and improves the behavior of the system.

A number of comments are in order:

– Feeding back the information from the state x of the system is expected to be, and it is, an effective way to alter the system behavior. This is because knowledge of the (initial) state and the input uniquely determines the system's future behavior and intuitively using the state information should be a good way to control the system, i.e., modifying its behavior.

– In a state feedback control law, the input u can be any function of the state $u = f(x, r)$, not necessarily linear with constant gain F as in (2). Typically given (1) and (2) is selected as the linear state feedback primarily because the resulting closed-loop description (3) is

also a linear time-invariant system. However, depending on the application needs, the state feedback control law (2) can be more complex.

– Although the Eqs. (3) that describe the closed-loop behavior are different from Eq. (1), this does not imply that the system parameters have changed. The way feedback control acts is not by actually changing the system parameters A, B, C, D but by changing u so that closed-loop system behaves as if the parameters were changed. When one applies, say, a step via $r(t)$ in the closed-loop system, then $u(t)$ in (2) is modified appropriately so the system behaves in a desired way.

– It is possible to implement u in (2) as an open-loop control law, namely,

$$\begin{aligned}\hat{u}(s) = {} &F[sI - (A + BF)]^{-1}x(0) \\ &+ [I - F(sI - A)^{-1}B]^{-1}\hat{r}(s) \quad (4)\end{aligned}$$

where Laplace transforms have been used for notational convenience. Equation (4) produces exactly the same input as Eq. (2), but it has the serious disadvantage that it is based exclusively on prior knowledge on the system (notably $x(0)$ and parameters A, B). As a result, when there are uncertainties (and there always are), the open-loop control law (4) may fail, while the closed-loop control law (2) succeeds.

– Analogous definitions exist for continuous-time, time-varying systems described by the equations

$$\dot{x} = A(t)x + B(t)u, \quad y = C(t)x + D(t)u \qquad (5)$$

In this framework, the control law is described by

$$u = F(t)x + r, \qquad (6)$$

and the resulting closed-loop system is

$$\begin{aligned}\dot{x} &= [A(t) + B(t)F(t)]x + B(t)r, \\ y &= [C(t) + D(t)F(t)]x + D(t)r.\end{aligned} \qquad (7)$$

Linear Discrete-Time Systems

For the discrete-time, time-invariant case, the system description is

$$x(k+1) = Ax(k) + Bu(k), \quad y = Cx(k) + Du(k), \tag{8}$$

the linear state feedback control law is defined as

$$u(k) = Fx(k) + r(k), \tag{9}$$

and the closed-loop system is described by

$$\begin{aligned} x(k+1) &= (A + BF)x(k) + Br(k), \\ y(k) &= (C + DF)x(k) + Dr(k). \end{aligned} \tag{10}$$

Similarly, for the discrete-time, time-varying case

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k), \end{aligned} \tag{11}$$

the control law is defined as

$$u(k) = F(k)x(k) + r(k), \tag{12}$$

and the resulting closed-loop system is

$$\begin{aligned} x(k+1) &= [A(k) + B(k)F(k)]x(k) + B(k)r(k), \\ y(k) &= [C(k) + D(k)F(k)]x(k) + D(k)r(k). \end{aligned} \tag{13}$$

Selecting the Gain F

F (or $F(t)$) is selected so that the closed-loop system has certain desirable properties. Stability is of course of major importance. Many control problems are addressed using linear state feedback including tracking and regulation, diagonal decoupling, and disturbance rejection. Here we shall focus on stability. Stability can be achieved under appropriate controllability assumptions. In the time-varying case, one way to determine such stabilizing $F(t)$ (or $F(k)$) is to use results from the optimal linear quadratic regulator (LQR)

theory which yields the "best" $F(t)$ (or $F(k)$) in some sense.

In the time-invariant case, one can also use a LQR formulation, but here stabilization is equivalent to the problem of assigning the n eigenvalues of $(A + BF)$ in the stable region of the complex plane. If $\lambda_i, i = 1, \ldots, n$, are the eigenvalues of $A + BF$, then F should be chosen so that, for all $i = 1, \ldots, n$, the real part of λ_i, $Re(\lambda_i) < 0$ in the continuous-time case, and the magnitude of λ_i, $|\lambda_i| < 1$ in the discrete-time case. Eigenvalue assignment is therefore an important problem, which is discussed hereafter.

Eigenvalue Assignment Problem

For continuous-time and discrete-time, time-invariant systems, the eigenvalue assignment problem can be posed as follows. Given matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$, find $F \in \mathbb{R}^{m \times n}$ such that the eigenvalues of $A + BF$ are assigned to arbitrary, complex conjugate, locations. Note that the characteristic polynomial of $A + BF$, namely, $\det(sI - (A + BF))$, has real coefficients, which implies that any complex eigenvalue is part of a pair of complex conjugate eigenvalues.

Theorem 1 *The eigenvalue assignment problem has a solution if and only if the pair (A, B) is reachable.*

For single-input systems, that is, for systems with $m = 1$, the eigenvalue assignment problem has a simple solution, as illustrated in the following statement:

Proposition 1 *Consider system (1) or (8). Let $m = 1$. Assume that*

$$rank\ R = n,$$

where

$$R = [B, AB, \ldots A^{n-1}B],$$

that is, the system is reachable. Let $p(s)$ be a desired monic polynomial of degree n. Then there is a (unique) linear state feedback gain matrix F such that the characteristic polynomial of $A + BF$ is equal to $p(s)$. Such linear state feedback gain matrix F is given by

$$F = -\begin{bmatrix} 0 \cdots 0\ 1 \end{bmatrix} R^{-1} p(A). \qquad (14)$$

Proposition 1 provides a constructive way to assign the characteristic polynomial, hence the eigenvalues, of the matrix $A + BF$. Note that, for low order systems, i.e., if $n = 2$ or $n = 3$, it may be convenient to compute directly the characteristic polynomial of $A + BF$ and then compute F using the principle of identity of polynomials, i.e., F should be such that the coefficients of the polynomials $\det(sI - (A + BF))$ and $p(s)$ coincide. Equation (14) is known as Ackermann's formula.

The result summarized in Proposition 1 can be extended to multi-input systems.

Proposition 2 *Consider system (1) or (8). Suppose*

$$rank\ R = n,$$

that is, the system is reachable. Let $p(s)$ be a monic polynomial of degree n. Then there is a linear state feedback gain matrix F such that the characteristic polynomial of $A + BF$ is equal to $p(s)$.

Note that in the case $m > 1$ the linear state feedback gain matrix F assigning the characteristic polynomial of the matrix $A + BF$ is not unique. To compute such a gain matrix, one may exploit the following fact:

Lemma 1 *Consider system (1). Suppose*

$$rank\ R = n,$$

that is, the system is reachable. Let b_i be a nonzero column of the matrix B. Then there is a matrix G such that the single-input system

$$\dot{x} = (A + BG)x + b_i v \qquad (15)$$

is reachable. Similar results are true for discrete-time systems (8).

Exploiting Lemma 1, it is possible to design a matrix F such that the characteristic polynomial of $A + BF$ equals some monic polynomial $p(s)$ of degree n in two steps. First, we compute a matrix G such that the system (15) is reachable, and

then we use Ackermann's formula to compute a linear state feedback gain matrix F such that the characteristic polynomial of

$$A + BG + b_i F$$

is $p(s)$. Note also that if (A, B) is reachable, under mild conditions on A, there exists vector g so that (A, Bg) is reachable.

There are many other methods to assign the eigenvalues which may be found in the references below.

Transfer Functions

If $H_F(s)$ is the transfer function matrix of the closed-loop system (3), it is of interest to find its relation to the open-loop transfer function $H(s)$ of (1). It can be shown that

$$\begin{aligned} H_F(s) &= H(s)[I - F(sI - A)^{-1}B]^{-1} \\ &= H(s)[F(sI - (A + BF))^{-1}B + I] \end{aligned}$$

In the single-input, single-output case, it can be readily shown that the linear state feedback control law (2) only changes the coefficients of the denominator polynomial in the transfer function (this result is also true in the multi-input, multi-output case). Therefore, if any of the (stable) zeros of $H(s)$ need to be changed, the only way to accomplish this via linear state feedback is by pole-zero cancelation (assigning closed-loop poles at the open-loop zero locations; in the MIMO case, closed-loop eigenvalue directions also need to be assigned for cancelations to take place). Note that it is impossible to change the unstable zeros of $H(s)$ under stability, since they would have to be canceled with unstable poles.

Observer-Based Dynamic Controllers

When the state x is not available for feedback, an asymptotic estimator (a Luenberger observer) is typically used to estimate the state. The estimate

$\tilde{x}$ of the state, instead of the actual x, is then used in (2) to control the system, in what is known as the certainty equivalence architecture.

Summary

The notion of state feedback for linear systems has been discussed. It has been shown that state feedback modifies the closed-loop behavior. The related problem of eigenvalue assignment has been discussed, and its connection with the reachability (controllability) properties of the system has been highlighted. The class of feedback laws considered is the simplest possible one. If additional constraints on the input signal, or on the closed-loop performance, are imposed, then one perhaps has to resort to nonlinear state feedback, for example, if the input signal is bounded in amplitude or rate. If constraints such as decoupling of the systems into m noninteracting subsystems or tracking under asymptotic stability are imposed, then dynamic state feedback may be necessary.

Cross-References

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
Chen CT (1984) Linear system theory and design. Holt, Rinehart and Winston, New York
DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs
Hespanha JP (2009) Linear systems theory. Princeton Press, Princeton
Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Wonham WM (1967) On pole assignment in multi-input controllable linear systems. IEEE Trans Autom Control AC-12:660–665
Wonham WM (1985) Linear multivariable control: a geometric approach, 3rd edn. Springer, New York

Linear Systems: Continuous-Time Impulse Response Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

An important input–output description of a linear continuous-time system is its impulse response, which is the response $h(t, \tau)$ to an impulse applied at time τ. In time-invariant systems that are also causal and at rest at time zero, the impulse response is $h(t, 0)$ and its Laplace transform is the transfer function of the system. Expressions for $h(t, \tau)$ when the system is described by state-variable equations are also derived.

Keywords

Continuous-time; Impulse response descriptions; Linear systems; Time-invariant; Time-varying; Transfer function descriptions

Introduction

Consider linear continuous-time dynamical systems, the input–output behavior of which can be described by an integral representation of the form

$$y(t) = \int_{-\infty}^{+\infty} H(t, \tau)u(\tau)d\tau \qquad (1)$$

where $t, \tau \in \mathbb{R}$, the output is $y(t) \in \mathbb{R}^p$, the input is $u(t) \in \mathbb{R}^m$, and $H : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{p \times m}$ is assumed to be integrable. For instance, any system in state-variable form

$$\dot{x} = A(t)x + B(t)u$$
$$y = C(t)x + D(t)u \tag{2}$$

or

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du \tag{3}$$

also has a representation of the form (1) as we shall see below.

Note that it is assumed that at $\tau = -\infty$, the system is at rest. $H(t, \tau)$ is the *impulse response matrix* of the system (1). To explain, consider first a single-input single-output system:

$$y(t) = \int_{-\infty}^{+\infty} h(t, \tau)u(\tau)d\tau, \tag{4}$$

and recall that if $\delta(\hat{t}-\tau)$ denotes an impulse (delta or Dirac) function applied at time $\tau = \hat{t}$, then for a function $f(t)$,

$$f(\hat{t}) = \int_{-\infty}^{+\infty} f(\tau)\delta(\hat{t} - \tau)d\tau. \tag{5}$$

If now in (4) $u(\tau) = \delta(\hat{t} - \tau)$, that is, an impulse input is applied at $\tau = \hat{t}$, then the output $y_I(t)$ is

$$y_I(t) = h(t, \hat{t}),$$

i.e., $h(t, \hat{t})$ is the output at time t when an impulse is applied at the input at time $\hat{t}$. So in (4), $h(t, \tau)$ is the response at time t to an impulse applied at time τ. Clearly if the *impulse response $h(t, \tau)$* is known, the response to any input $u(t)$ can be derived via (4), and so $h(t, \tau)$ is an input/output description of the system.

Equation (1) is a generalization of (4) for the multi-input, multi-output case. If we let all the components of $u(\tau)$ in (1) be zero except the jth component, then

$$y_i(t) = \int_{-\infty}^{+\infty} h_{ij}(t, \tau)u_j(\tau)d\tau, \tag{6}$$

$h_{ij}(t, \tau)$ denotes the response of the ith component of the output of system (1) at time t due to an impulse applied to the jth component of the

input at time τ with all remaining components of the input being zero. $H(t, \tau) = [h_{ij}(t, \tau)]$ is called the *impulse response matrix* of the system.

If it is known that system (1) is causal, then the output will be zero before an input is applied. Therefore,

$$H(t, \tau) = 0, \quad \text{for } t < \tau, \tag{7}$$

and (1) becomes

$$y(t) = \int_{-\infty}^{t} H(t, \tau)u(\tau)d\tau. \tag{8}$$

Rewrite (8) as

$$y(t) = \int_{-\infty}^{t_0} H(t, \tau)u(\tau)d\tau + \int_{t_0}^{t} H(t, \tau)u(\tau)d\tau$$

$$= y(t_0) + \int_{t_0}^{t} H(t, \tau)u(\tau)d\tau. \tag{9}$$

If (1) is at rest at $t = t_0$ (i.e., if $u(t) = 0$ for $t \geq t_0$, then $y(t) = 0$ for $t \geq t_0$), $y(t_0) = 0$ and (9) becomes

$$y(t) = \int_{t_0}^{t} H(t, \tau)u(\tau)d\tau. \tag{10}$$

If in addition system (1) is time-invariant, then $H(t, \tau) = H(t - \tau, 0)$ (also written as $H(t - \tau)$) since only the elapsed time $(t - \tau)$ from the application of the impulse is important. Then (10) becomes

$$y(t) = \int_{0}^{t} H(t - \tau)u(\tau)d\tau, \quad t \geq 0, \tag{11}$$

where we chose $t_0 = 0$ without loss of generality. Equation (11) is the description for causal, time-invariant systems, at rest at $t = 0$.

Equation (11) is a convolution integral and if we take the (one-sided or unilateral) Laplace transform of both sides,

$$\hat{y}(s) = \hat{H}(s)\hat{u}(s), \tag{12}$$

where $\hat{y}(s)$, $\hat{u}(s)$ are the Laplace transforms of $y(t)$, $u(t)$ and $\hat{H}(s)$ is the Laplace transform of

the impulse response $H(t)$. $\hat{H}(s)$ is the *transfer function matrix* of the system. Note that the transfer function of a linear, time-invariant system is typically defined as the rational matrix $\hat{H}(s)$ that satisfies (12) for any input and its corresponding output assuming zero initial conditions, which is of course consistent with the above analysis.

Connection to State-Variable Descriptions

When a system is described by the state-variable description (2), then

$$y(t) = \int_{t_0}^{t} [C(t)\Phi(t, \tau)B(\tau) + D(t)\delta(t - \tau)]u(\tau)d\tau, \quad (13)$$

where it was assumed that $x(t_0) = 0$, i.e., the system is at rest at t_0. Here $\Phi(t, \tau)$ is the state transition matrix of the system defined by the Peano-Baker series:

$$\Phi(t, t_0) = I + \int_{t_0}^{t} A(\tau_1)d\tau_1$$
$$+ \int_{t_0}^{t} A(\tau_1) \left[\int_{t_0}^{\tau_1} A(\tau_2)d\tau_2\right] d\tau_1 + \cdots;$$

see ▶ Linear Systems: Continuous-Time, Time–Varying State Variable Descriptions.

Comparing (13) with (10), the impulse response

$$H(t, \tau) = \begin{cases} C(t)\Phi(t, \tau)B(t) + D(t)\delta(t-\tau) & t \geq \tau, \\ 0 & t < \tau. \end{cases} \quad (14)$$

Similarly, when the system is time-invariant and is described by (3),

$$y(t) = \int_{t_0}^{t} [Ce^{A(t-\tau)}B + D\delta(t - \tau)]u(\tau)d\tau, \quad (15)$$

where $x(t_0) = 0$.

Comparing (15) with (11), the impulse response

$$H(t - \tau) = \begin{cases} Ce^{A(t-\tau)}B + D\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau, \end{cases} \quad (16)$$

or as it is commonly written (taking the time when the impulse is applied to be zero, $\tau = 0$)

$$H(t) = \begin{cases} Ce^{At}B + D\delta(t) & t \geq 0, \\ 0 & t < 0. \end{cases} \quad (17)$$

Take now the (one-sided or unilateral) Laplace transform of both sides in (17) to obtain

$$\hat{H}(s) = C(sI - A)^{-1}B + D, \quad (18)$$

which is the transfer function matrix in terms of the coefficient matrices in the state-variable description (3). Note that (18) can also be derived directly from (3) by assuming zero initial conditions ($x(0) = 0$) and taking Laplace transform of both sides.

Finally, it is easy to show that equivalent state-variable descriptions give rise to the same impulse responses.

Summary

The continuous-time impulse response is an external, input–output description of linear, continuous-time systems. When the system is time-invariant, the Laplace transform of the impulse response $h(t, 0)$ (which is the output response at time t due to an impulse applied at time zero with initial conditions taken to be zero) is the transfer function of the system – another very common input–output description. The relationships with the state-variable descriptions are shown.

Cross-References

Recommended Reading

External or input–output descriptions such as the impulse response and the transfer function (in the time-invariant case) are described in several textbooks below.

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs
Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs
Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York

Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Synonyms

LTI Systems

Abstract

Continuous-time processes that can be modeled by linear differential equations with constant coefficients can also be described in a systematic way in terms of state variable descriptions of the form $\dot{x}(t) = Ax(t) + Bu(t)$, $y(t) = Cx(t) + Du(t)$. The response of such systems due to a given input and a set of initial conditions is derived and expressed in terms of the variation of constants formula. Equivalence of state variable descriptions is also discussed.

Keywords

Continuous-time; Linear systems; State variable descriptions; Time-invariant

Introduction

Linear, continuous-time systems are of great interest because they model, exactly or approximately, the behavior over time of many practical physical systems of interest. We are particularly interested in systems, the behavior of which is described by linear, ordinary differential equations with constant coefficients.

Such descriptions can always be rewritten as a set of first-order differential equations, typically in the following convenient state variable form:

$$\dot{x} = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t);$$
$$x(0) = x_0, \tag{1}$$

where $x(t)$, the state vector, is a column vector of dimension n ($x(t) \in \mathbb{R}^n$) and $\dot{x}(t) = \frac{dx}{dt}$ with the derivative being taken element by element. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are matrices with real entries (these are the constant coefficients that make the system time invariant); and $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ are the inputs and outputs of the system. The vector differential equation is the *state equation* and the algebraic equation is the *output equation*.

The advantage of the above state variable description is that for given input $u(t)$ and initial condition $x(0)$, its solution (state and output motions or trajectories) can be conveniently and systematically characterized. This is shown below.

Deriving State Variable Descriptions

Description (1) may be derived directly, by modeling the behavior of a linear, continuous-time, time-invariant system, but more often it is derived either from the linearization of a nonlinear equation around an operating point or a trajectory or from higher-order differential equations that model the system's behavior. The example below illustrates the latter case.
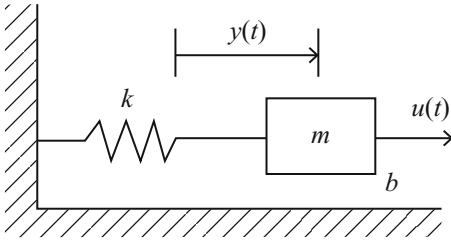
Consider a spring-mass example, where a mass m slides horizontally on a surface with damping coefficient b due to friction and it is attached to a wall by a linear spring of spring constant k. If $y(t)$ denotes the distance of the center of the mass from a position of rest of the spring, by applying Newton's law the following second-order linear ordinary differential equation with constant coefficients is obtained:

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = u(t). \qquad (2)$$

Here $\dot{y}(t) = \frac{dy(t)}{dt}$. The motion of the mass $y(t), t > 0$ is uniquely determined if the applied force $u(t), t \geq 0$ is known and at $t = 0$ the initial position $y(0) = y_0$ and initial velocity $\dot{y}(0) = y_1$ are given. To obtain a state variable description, introduce the state variables x_1 and x_2 as

$$x_1(t) = y(t), \quad x_2(t) = \dot{y}(t)$$

to obtain the set of first-order differential equations $m\dot{x}_2(t) + b\dot{x}_2(t) + kx_1(t) = u(t)$ and $\dot{x}(t) = x_2(t)$ which can be rewritten in the form of (1)

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) \qquad (3)$$

and

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

with $\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}$ as initial conditions. This is of the form (1) where $x(t)$ is a 2-dimensional column vector; A is a 2×2 matrix; B and C are 2-dimensional column and row vectors, respectively; and $x(0) = x_0$.

Notes:

1. It is always possible to obtain a state variable description which is equivalent to a given set of higher-order differential equations

2. The choice of the state variables, here x_1 and x_2, is not unique. Different choices will lead to different A, B, and C.

3. The number of the state variables is typically equal to the order of the set of the higher-order differential equations and equals the number of initial conditions needed to derive the unique solution; in the above example this number is 2.

4. In time-invariant systems, it can be assumed without loss of generality that the starting time is $t = 0$ and so the initial conditions are taken to be $x(0) = x_0$.

Solving $\dot{x} = A(t)x; \; x(0) = x_0$

Consider the homogeneous equation

$$\dot{x} = A(t)x; \quad x(0) = x_0 \qquad (4)$$

where $x(t) = [x_1(t), \ldots, x_n(t)]^T$ is the state vector of dimension n and A is an $n \times n$ matrix with entries real numbers (i.e., $A \in \mathbb{R}^{n \times n}$).

Equation (4) is a special case of (1) where there are no inputs and outputs, u and y. The homogeneous vector differential equation (4) will be solved first, and its solution will be used to find the solution of (1).

Solving (4) is an initial value problem. It can be shown that there always exists a unique solution $\varphi(t)$ such that

$$\dot{\varphi}(t) = A\varphi(t); \quad \varphi(0) = x_0.$$

To find the unique solution, consider first the one-dimensional case, namely,

$$\dot{y}(t) = ay(t); \quad y(0) = y_0$$

the unique solution of which is

$$y(t) = e^{at}y_0, \quad t \geq 0.$$

The scalar exponential e^{at} can be expressed in a series form

$$e^{at} = \sum_{k=0}^{\infty} \frac{t^k}{k!} a^k \left(= 1 + \frac{1}{1}ta + \frac{1}{2}t^2a^2 + \frac{1}{6}t^3a^3 + \dots\right)$$

The generalization to the $n \times n$ matrix exponential (A is $n \times n$) is given by

$$e^{At} = \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k \quad \left(= I_n + At + \frac{1}{2}A^2t^2 + \dots\right) \tag{5}$$

By analogy, let the solution to (4) be

$$(x(t) =)\varphi(t) = e^{At}x_0 \tag{6}$$

It is a solution since if it is substituted into (4),

$$\dot{\varphi}(t) = [A + At + \frac{1}{2}A^2t^2 + \dots]x_0$$
$$= Ae^{At}x_0 = A\varphi(t)$$

and $\varphi(0) = e^{A \cdot 0}x_0 = x_0$, that is, it satisfies the equation and the initial condition. Since the solution of (4) is unique, (6) is the unique solution of (4).

The solution (6) can be derived more formally using the Peano-Baker series (see ▸ Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions), which in the present time-invariant case becomes the defining series for the matrix exponential (5).

System Response

Based on the solution of the homogeneous equation (4), shown in (6), the solution of the state equation in (1) can be shown to be

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \tag{7}$$

The following properties for the matrix exponential e^{At} can be shown directly from the defining series:
1. $Ae^{At} = e^{At}A$.
2. $(e^{At})^{-1} = e^{-At}$.

Equation (7) which is known as the *variation of constants formula* can be derived as follows:

Consider $\dot{x} = Ax + Bu$ and let $z(t) = e^{-At}x(t)$. Then $x(t) = e^{At}z(t)$ and substituting

$$Ae^{At}z(t) + e^{At}\dot{z}(t) = Ae^{At}z(t) + Bu(t)$$

or $\dot{z}(t) = e^{-At}Bu(t)$ from which

$$z(t) - z(0) = \int_0^t e^{-A\tau}Bu(\tau)d\tau$$

or

$$e^{-At}x(t) - x(0) = \int_0^t e^{-A\tau}Bu(\tau)d\tau$$

or

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

which is the variation of constants formula (7).

Equation (7) is the sum of two parts, the *state response* (when $u(t) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x_0 = 0$ and the system is driven only by the input $u(t)$); this illustrates the linear system principle of superposition.

If the output equation $y(t) = Cx(t) + Du(t)$ is considered, then in view of (7),

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t)$$
$$\tag{8}$$

$$= Ce^{At}x_0 + \int_0^t [Ce^{A(t-\tau)}B$$

$$+ D\delta(t-\tau)]u(\tau)d\tau$$

The second expression involves the Dirac (or impulse or delta) function $\delta(t)$, and it is derived based on the basic property for $\delta(t)$, namely,

$$f(t) = \int_{-\infty}^{\infty} \delta(t-\tau)f(\tau)d\tau$$

It is clear that the matrix exponential e^{At} plays a central role in determining the response of a linear continuous-time, time-invariant system described by (1).

Given A, e^{At} may be determined using several methods including its defining series, diagonalization of A using a similarity transformation (PAP^{-1}), the Cayley-Hamilton theorem, using expressions involving the modes of the system ($e^{At} = \sum_{i=1}^n A_i e^{\lambda_i t}$ when A has n distinct eigenvalues λ_i; $A_i = v_i \tilde{v}_i$ with v_i, $\tilde{v}_i$ the right and left eigenvectors of A that correspond to λ_i ($\tilde{v}_i v_j = 1$, $i = j$ and $\tilde{v}_i v_j = 0$, $i \neq j$)), or using Laplace transform ($e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$). See references below for detailed algorithms.

Equivalent State Variable Descriptions

Given

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \tag{9}$$

consider the new state vector $\tilde{x}$ where

$$\tilde{x} = Px$$

with P a real nonsingular matrix. Substituting $x = P^{-1}\tilde{x}$ in (9), we obtain

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u, \quad y = \tilde{C}\tilde{x} + \tilde{D}u, \tag{10}$$

where

$$\tilde{A} = PAP^{-1}, \ \tilde{B} = PB, \ \tilde{C} = CP^{-1}, \ \tilde{D} = D$$

The state variable descriptions (9) and (10) are called equivalent and P is the equivalence transformation. This transformation corresponds to a change in the basis of the state space, which is a vector space. Appropriately selecting P, one can simplify the structure of $\tilde{A}(= PAP^{-1})$; the matrices $\tilde{A}$ and A are called similar. When the eigenvectors of A are all linearly independent (this is the case, e.g., when all eigenvalues λ_i of A are distinct), then P may be found so that $\tilde{A}$ is diagonal. When e^{At} is to be determined, and $\tilde{A} = PAP^{-1} = \text{diag}[\lambda_i]$ ($\tilde{A}$ and A have the same eigenvalues), then

$$e^{At} = e^{P^{-1}\tilde{A}Pt} = P^{-1}e^{\tilde{A}t}P = P^{-1}\text{diag}[e^{\lambda_i t}]P.$$

Note that it can be easily shown that equivalent state space representations give rise to the same impulse response and transfer function (see ▶ Linear Systems: Continuous-Time Impulse Response Descriptions).

Summary

State variable descriptions for continuous-time time-invariant systems are introduced and the state and output responses to inputs and initial conditions are derived. Equivalence of state variable representations is also discussed.

Cross-References

▶ Linear Systems: Continuous-Time Impulse Response Descriptions

▶ Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions

▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Recommended Reading

The state variable description of systems received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman and others in filtering theory and quadratic control theory and to the work of applied mathematicians concerned with the stability theory of dynamical systems. For comments and extensive references on some of the early contributions in these areas, see Kailath (1980) and Sontag (1990). The use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Brockett RW (1970) Finite dimensional linear systems. Wiley, New York

Chen CT (1984) Linear system theory and design. Holt, Rinehart and Winston, New York

DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs

Hespanha JP (2009) Linear systems theory. Princeton Press, Princeton

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York

Zadeh LA, Desoer CA (1963) Linear system theory: the state space approach. McGraw-Hill, New York

Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Continuous-time processes that can be modeled by linear differential equations with time-varying coefficients can be written in terms of state variable descriptions of the form $\dot{x}(t) = A(t)x(t) + B(t)u(t)$, $y(t) = C(t)x(t) + D(t)u(t)$. The response of such systems due to a given input and initial conditions is derived using the Peano-Baker series. Equivalence of state variable descriptions is also discussed.

Keywords

Introduction

Dynamical processes that can be described or approximated by linear high-order ordinary differential equations with time-varying coefficients can also be described, via a change of variables, by state variable descriptions of the form

$$\dot{x}(t) = A(t)x(t) + B(t)u(t); \quad x(t_0) = x_0$$
$$y(t) = C(t)x(t) + D(t)u(t), \tag{1}$$

where $x(t)$ ($t \in \mathbb{R}$, the set of reals) is a column vector of dimension n ($x(t) \in \mathbb{R}^n$) and $A(t), B(t), C(t), D(t)$ are matrices with entries functions of time t. $A(t) = [a_{ij}(t)]$, $a_{ij}(t) : \mathbb{R} \to \mathbb{R}$. $A(t) \in \mathbb{R}^{n \times n}, B(t) \in \mathbb{R}^{n \times m}, C(t) \in \mathbb{R}^{p \times n}, D(t) \in \mathbb{R}^{p \times m}$. The input vector is $u(t) \in \mathbb{R}^m$ and the output vector is $y(t) \in \mathbb{R}^p$. The vector differential equation in (1) is the *state equation*, while the algebraic equation is the *output equation*.

The advantage of the state variable description (1) is that given an input $u(t)$, $t \geq 0$ and an initial condition $x(t_0) = x_0$, the state trajectory or motion for $t \geq t_0$ can be conveniently characterized. To derive the expressions, we first consider the homogenous state equation and the corresponding initial value problem.

Solving $\dot{x}(t) = A(t)x(t); \; x(t_0) = x_0$

Consider the homogenous equation with the initial condition

$$x(t) = A(t)x(t); \quad x(t_0) = x_0 \qquad (2)$$

where $x(t) = [x_1(t), \ldots, x_n(t)]^T$ is the state (column) vector of dimension n and $A(t)$ is an $n \times n$ matrix with entries functions of time that take on values from the field of reals ($A \in \mathbb{R}^{n \times n}$).

Under certain assumptions on the entries of $A(t)$, a solution of (2) exists and it is unique. These assumptions are satisfied, and a solution exists and is unique in the case, for example, when the entries of $A(t)$ are continuous functions of time. In the following we make this assumption.

To find the unique solution of (2), we use the *method of successive approximations* which when applied to

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0 \qquad (3)$$

is described by

$$\phi_0(t) = x_0$$

$$\phi_m(t) = x_0 + \int_{t_0}^{t} f(\tau, \phi_{m-1}(\tau)) d\tau, \quad m = 1, 2, \ldots \qquad (4)$$

As $m \to \infty$, ϕ_m converges to the unique solution of (3), assuming the f satisfies certain conditions.

Applying the method of successive approximations to (2) yields

$$\phi_0(t) = x_0$$

$$\phi_1(t) = x_0 + \int_{t_0}^{t} A(\tau)x_0 d\tau$$

$$\phi_2(t) = x_0 + \int_{t_0}^{t} A(\tau)\phi_1(\tau)x_0 d\tau$$

$$\vdots$$

$$\phi_m(t) = x_0 + \int_{t_0}^{t} A(\tau)\phi_{m-1}(\tau)x_0 d\tau$$

from which

$$\phi_m(t) = \left[I + \int_{t_0}^{t} A(\tau_1) d\tau_1 \right.$$

$$+ \int_{t_0}^{t} A(\tau_1) \int_{t_0}^{\tau_1} A(\tau_2) d\tau_2 d\tau_1 + \ldots$$

$$+ \int_{t_0}^{t} A(\tau_1) \int_{t_0}^{\tau_1} A(\tau_2) \ldots \int_{t_0}^{\tau_{m-1}} A(\tau_m)$$

$$\left. d\tau_m \ldots d\tau_1 \right] x_0$$

When $m \to \infty$, and under the above continuity assumptions on $A(t)$, $\phi_m(t)$ converges to the unique solution of (2), i.e.,

$$\phi(t) = \Phi(t, t_0)x_0 \qquad (5)$$

where

$$\Phi(t, t_0) = I + \int_{t_0}^{t} A(\tau_1) d\tau_1$$

$$+ \int_{t_0}^{t} A(\tau_1) \left[\int_{t_0}^{\tau_1} A(\tau_2) d\tau_2 \right] d\tau_1 + \ldots \qquad (6)$$

Note that $\Phi(t_0, t_0) = I$ and by differentiation it can be seen that

$$\dot{\phi}(t, t_0) = A(t)\phi(t, t_0), \qquad (7)$$

as expected, since (5) is the solution of (2). The $n \times n$ matrix $\Phi(t, t_0)$ is called the *state transition matrix* of (2). The defining series (6) is called the Peano-Baker series.

Note that when $A(t) = A$, a constant matrix, then (6) becomes

$$\Phi(t, t_0) = I + \sum_{k=1}^{\infty} \frac{A^k (t - t_0)^k}{k!} \qquad (8)$$

L

which is the defining series for the matrix exponential $e^{A(t-t_0)}$ (see ► Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions).

System Response

Based on the solution (5) of $\dot{x} = A(t)x(t)$, the solution of the non-homogenous equation

$$\dot{x}(t) = A(t)x(t) + B(t)u(t); \quad x(t_0) = x_0 \quad (9)$$

can be shown to be

$$\phi(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau. \quad (10)$$

Equation (10) is the *variation of constants formula*. This result can be shown via direct substitution of (10) into (9); note that $\phi(t) = \Phi(t_0, t_0)x_0 = x_0$. That (10) is a solution can also be shown using a change of variables in (9), namely,

$$z(t) = \Phi(t_0, t)x(t).$$

Equation (10) is the sum of two parts, the *state response* (when $u(t) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x_0 = 0$ and the system is driven only by the input $u(t)$); this illustrates the linear system principle of superposition.

In view of (10), the output $y(t) (= C(t)x(t) + D(t)u(t))$ is

$$y(t) = C(t)\Phi(t, t_0)x_0$$

$$+ \int_{t_0}^{t} C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau + D(t)u(t)$$

$$= C(t)\Phi(t, t_0)x_0 + \int_{t_0}^{t} [C(t)\Phi(t, \tau)B(\tau)$$

$$+ D(t)\delta(t - \tau)]u(\tau)d\tau$$

The second expression involves the Dirac (or impulse or delta) function $\delta(t)$ and is derived based on the basic property for $\delta(t)$, namely,

$$f(t) = \int_{-\infty}^{+\infty} \delta(t - \tau)f(\tau)d\tau,$$

where $\delta(t - \tau)$ denotes an impulse applied at time $\tau = t$.

Properties of the State Transition Matrix $\Phi(t, t_0)$

In general it is difficult to determine $\Phi(t, t_0)$ explicitly; however, $\Phi(t, t_0)$ may be readily determined in a number of special cases including the cases in which $A(t) = A$, $A(t)$ is diagonal, $A(t)A(\tau) = A(\tau)A(t)$.

Consider $\dot{x} = A(t)x$. We can derive a number of important properties which are described below. It can be shown that given n linearly independent initial conditions x_{0i}, the corresponding n solutions $\phi_i(t)$ are also linearly independent. Let a *fundamental matrix* $\Psi(t)$ of $\dot{x} = A(t)x$ be an $n \times n$ matrix, the columns of which are a set of linearly independent solutions $\phi_1(t), \ldots, \phi_n(t)$. The state transition matrix Φ is the fundamental matrix determined from solutions that correspond to the initial conditions $[1, 0, 0, \ldots]^T$, $[0, 1, 0, \ldots, 0]^T, \ldots [0, 0, \ldots, 1]^T$ (recall that $\Phi(t_0, t_0) = I$). The following are properties of $\Phi(t, t_0)$:

(i) $\Phi(t, t_0) = \Psi(t)\Psi^{-1}(t_0)$ with $\Psi(t)$ any fundamental matrix.

(ii) $\Phi(t, t_0)$ is nonsingular for all t and t_0.

(iii) $\Phi(t, \tau) = \Phi(t, \sigma)\Phi(\sigma, \tau)$ (semigroup property).

(iv) $[\Phi(t, t_0)]^{-1} = \Phi(t_0, t)$.

In the special case of time-invariant systems and $\dot{x} = Ax$, the above properties can be written in terms of the matrix exponential since

$$\Phi(t, t_0) = e^{A(t-t_0)}.$$

Equivalence of State Variable Descriptions

Given the system

$$\dot{x} = A(t)x + B(t)u$$
$$y = C(t)x + D(t)u \tag{11}$$

consider the new state vector $\tilde{x}$

$$\tilde{x}(t) = P(t)x(t)$$

where $P^{-1}(t)$ exists and P and P^{-1} are continuous. Then the system

$$\dot{\tilde{x}} = \tilde{A}(t)\tilde{x} + \tilde{B}(t)u$$
$$y = \tilde{C}(t)\tilde{x} + \tilde{D}(t)u$$

where

$$\tilde{A}(t) = [P(t)A(t) + \dot{P}(t)]P^{-1}(t)$$
$$\tilde{B}(t) = P(t)B(t)$$
$$\tilde{C}(t) = C(t)P^{-1}(t)$$
$$\tilde{D}(t) = D(t)$$

is equivalent to (1). It can be easily shown that equivalent descriptions give rise to the same impulse responses.

Summary

State variable descriptions for continuous-time time-varying systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

Recommended Reading

Additional information regarding the time-varying case may be found in Brockett (1970), Rugh (1996), and Antsaklis and Michel (2006). For historical comments and extensive references on some of the early contributions, see Sontag (1990) and Kailath (1980).

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Brockett RW (1970) Finite dimensional linear systems. Wiley, New York

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Miller RK, Michel AN (1982) Ordinary differential equations. Academic, New York

Rugh WJ (1996) Linear system theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Sontag ED (1990) Mathematical control theory: deterministic finite dimensional systems. Texts in applied mathematics, vol 6. Springer, New York

L

Linear Systems: Discrete-Time Impulse Response Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

An important input-output description of a linear discrete-time system is its (discrete-time) impulse response (or pulse response), which is the response $h(k, k_0)$ to a discrete impulse applied at time k_0. In time-invariant systems that are also causal and at rest at time zero, the impulse response is $h(k, 0)$, and its z-transform is the transfer function of the system. Expressions for $h(k, k_0)$ when the system is described by state variable equations are derived.

Keywords

At rest; Causal; Discrete-time; Discrete-time impulse response descriptions; Linear systems; Pulse response descriptions; Time-invariant; Time-varying; Transfer function descriptions

Introduction

Consider linear, discrete-time dynamical systems that can be described by

$$y(k) = \sum_{l=-\infty}^{+\infty} H(k,l)u(l) \tag{1}$$

where $k, l \in \mathbb{Z}$ is the set of integers, the output is $y(k) \in \mathbb{R}^p$, the input is $u(k) \in \mathbb{R}^m$, and $H(k,l) : \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}^{p \times m}$. For instance, any system that can be written in state variable form

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) + D(k)u(k) \end{aligned} \tag{2}$$

or

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{3}$$

can be represented by (1). Note that it is assumed that at $l = -\infty$, the system is at rest, i.e., no energy is stored in the system at time $-\infty$.

Define the discrete-time impulse (or unit pulse) as

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}, k \in \mathbb{Z}$$

and consider a single-input, single-output system:

$$y(k) = \sum_{l=-\infty}^{+\infty} h(k,l)u(l) \tag{4}$$

If $u(l) = \delta(\hat{l} - l)$, that is, the input is a unit pulse applied at $l = \hat{l}$, then the output is

$$y_I(k) = h(k, \hat{l}),$$

i.e., $h(k, \hat{l})$ is the output at time k when a unit pulse is applied at time $\hat{l}$.

So in (4) $h(k, l)$ is the response at time k to a discrete-time impulse (unit pulse) applied at time l. $h(k, l)$ is the *discrete-time impulse response* of the system. Clearly if $h(k, l)$ is known, the response of the system to any input can be determined via (4). So $h(k, l)$ is an input/output description of the system.

Equation (1) is a generalization of (4) for the multi-input, multi-output case. If we let all the components of $u(l)$ in (1) be zero except for the jth component, then

$$y_i(k) = \sum_{l=-\infty}^{+\infty} h_{ij}(k,l)u_j(l) \tag{5}$$

$h_{ij}(k, l)$ denotes the response of the ith component of the output of system (1) at time k due to a discrete impulse applied to the jth component of the input at time l with all remaining components of the input being zero. $H(k, l) = [h_{ij}(k, l)]$ is called the *impulse response matrix* of the system.

If it is known that system (1) is *causal*, then the output will be zero before an input is applied. Therefore,

$$H(k, l) = 0, \quad \text{for } k < l, \tag{6}$$

and so when causality is present, (1) becomes

$$y(k) = \sum_{l=-\infty}^{k} H(k,l)u(l). \tag{7}$$

A system described by (1) is at rest at $k = k_0$ if $u(k) = 0$ for $k \geqslant k_0$ implies $y(k) = 0$ for $k \geqslant k_0$. For a system at rest at $k = k_0$, (7) becomes

$$y(k) = \sum_{l=k_0}^{k} H(k,l)u(l). \tag{8}$$

If system (1) is time-invariant, then $H(k, l) = H(k - l, 0)$ (also written as $H(k - l)$) since only the time elapsed $(k - l)$ from the application of the discrete-time impulse is important. Then (8) becomes

$$y(k) = \sum_{l=0}^{k} H(k-l)u(l), \quad k \geq 0, \quad (9)$$

where we chose $k_0 = 0$ without loss of generality. Equation (9) is the description for casual, time-invariant systems, at rest at $k = 0$.

Equation (9) is a convolution sum and if we take the (one-sided or unilateral) z-transform of both sides,

$$\hat{y}(z) = \hat{H}(z)\hat{u}(z), \quad (10)$$

where $\hat{y}(z)$, $\hat{u}(z)$ are the z-transforms of $y(k)$, $u(k)$ and $\hat{H}(z)$ is the z-transform of the discrete-time impulse response $H(k)$. $\hat{H}(z)$ is the *transfer function matrix* of the system. Note that the transfer function of a linear, time-invariant system is defined as the rational matrix $\hat{H}(z)$ that satisfies (10) for any input and its corresponding output assuming zero initial conditions.

Connections to State Variable Descriptions

When a system is described by (2), then

$$y(k) = \sum_{l=k_0}^{k-1} C(k)\Phi(k, l+1)B(l)u(l)$$
$$+ D(k)u(k), \quad k > k_0 \quad (11)$$

where it was assumed that $x(k_0) = 0$, i.e., the system is at rest at k_0. Here $\Phi(k, l) (= A(k-1)\cdots A(l))$ is the state transition matrix of the system.

Comparing (11) with (8), the discrete-time impulse response of the system is

$$H(k, l) = \begin{cases} C(k)\Phi(k, l+1)B(l) & k > l \\ D(k) & k = l \\ 0 & k < l \end{cases} \quad (12)$$

Similarly, when the system is time-invariant and is described by (3),

$$y(k) = \sum_{l=k_0}^{k-1} C A^{k-(l+1)} Bu(l) + Du(k), \quad k > k_0 \quad (13)$$

where $x(k_0) = 0$ and

$$H(k, l) = H(k-l) = \begin{cases} C A^{k-(l+1)} B & k > l \\ D & k = l \\ 0 & k < l \end{cases} \quad (14)$$

When $l = 0$ (taking the time when the discrete impulse is applied to be zero, $l = 0$), the discrete-time impulse response is

$$H(k) = \begin{cases} C A^{k-1} B & k > 0 \\ D & k = 0 \\ 0 & k < 0 \end{cases} \quad (15)$$

Taking (one-sided or unilateral) z-transforms of both sides in (15),

$$\hat{H}(z) = C(zI - A)^{-1}B + D \quad (16)$$

which is the transfer function matrix in terms of the coefficient matrices in the state variable description (3). Note that (16) can also be derived directly from (3) by assuming zero initial conditions ($x(0) = 0$) and taking z-transforms of both sides.

Finally, it is easy to show that equivalent state variable descriptions give rise to the same discrete-impulse response.

Summary

The discrete-time impulse response is an external, input-output description of linear, discrete-time systems. When the system is time-invariant, the z-transform of the impulse response $h(k, 0)$ (which is the output response at time k due to a discrete impulse applied at time zero with initial conditions taken to be zero) is the transfer function – another very common input-output description. The relationships to the state variable descriptions were shown.

Cross-References

Recommended Reading

External or input-output descriptions such as the impulse response and the transfer function (in the time-invariant case) are described in several textbooks below.

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston
Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs
Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Discrete-time processes that can be modeled by linear difference equations with constant coefficients can also be described in a systematic way in terms of state variable descriptions of the form $x(k + 1) = Ax(k) + Bu(k)$, $y(k) = Cx(k) + Du(k)$. The response of such systems due to a given input and subject to initial conditions is derived. Equivalence of state variable descriptions is also discussed.

Keywords

Introduction

Discrete-time systems arise in a variety of ways in the modeling process. There are systems that are inherently defined only at discrete points in time; examples include digital devices, inventory systems, economic systems such as banking where interest is calculated and added to savings accounts at discrete time interval, etc. There are also systems that describe continuous-time systems at discrete points in time; examples include simulations of continuous processes using digital computers and feedback control systems that employ digital controllers and give rise to sampled-data systems.

Linear, discrete-time, time-invariant systems can be modeled via state variable equations, namely,

$$x(k + 1) = Ax(k) + Bu(k); \quad x(0) = x_0$$
$$y(k) = Cx(k) + Du(k)$$

$$(1)$$

where $k \in \mathbb{Z}$, the set of integers, the state vector $x \in \mathbb{R}^n$, i.e., an n dimensional column vector; $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$ are matrices with entries of real numbers; and $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$ the output and the input, respectively. The vector difference equation in (1) is the *state equation* and the algebraic equation is the *output equation*.

Note that (1) could have been equivalently written as $x(l) = Ax(l - 1) + Bu(l - 1)$ where $l = k + 1$ and $x(l - 1)$ is an easily visualized delayed version of $x(l)$; this is a form more common in signal processing (where a two-sided or bilateral z-transform is used). In control where we assume a known initial condition at time equal to zero (and one-sided or unilateral z-transform is taken), the form in (1) is common.

Similar to the continuous-time case, (1) can be derived from a set of high-order difference equations by introducing the state variables

$x(k) = [x_1(k), \ldots, x_n(k)]^T$. Description (1) can also be derived from continuous-time system descriptions by sampling (see ▶ Sampled-Data Systems).

The advantage of the above state variable description is that given any input $u(k)$ and initial conditions $x(0)$, its solution (state trajectory or motion) can be conveniently and systematically characterized. This is done below. We first consider the solutions of the homogenous equation $x(k + 1) = Ax(k)$.

Solving $x(k + 1) = Ax(k)$; $x(0) = x_0$

Consider the homogenous equation

$$x(k + 1) = Ax(k); \quad x(0) = x_0 \qquad (2)$$

where $k \in \mathbb{Z}^+$ is a nonnegative integer, $x(k) = [x_1(k), \ldots, x_n(k)]^T$ is the state column vectors of dimension n, and A is an $n \times n$ matrix with entries real numbers (i.e., $A \in \mathbb{R}^{n \times n}$).

Write (2) for $k = 0, 1, 2, \ldots$, namely, $x(1) = Ax(0)$, $x(2) = Ax(1) = A^2x(0), \ldots$ to derive the solution

$$x(k) = A^k x(0), \quad k \geq 0 \qquad (3)$$

This result can be shown formally by induction. Note that $A^0 = I$ by convention and so (3) also satisfies the initial condition.

If the initial time were some (integer) k_0 instead of zero, then the solution would be

$$x(k) = A^{k-k_0} x(k_0), \quad k \geq k_0 \qquad (4)$$

The solution can be written as

$$x(k) = \Phi(k, k_0) x(k_0)$$
$$= \Phi(k - k_0, 0) x(k_0), \quad k \geq k_0 \qquad (5)$$

where $\Phi(k, k_0)$ is the state transition matrix and it equals $\Phi(k, k_0) = A^{k-k_0}$. Note that for time-invariant systems, the initial time k_0 can always be taken to be zero without loss of generality; this is because the behavior depends only on the

time elapsed $(k - k_0)$ and not on the actual initial time k_0.

In view of (3), it is clear that A^k plays an important role in the solutions of the difference state equations that describe linear, discrete-time, time-invariant systems; it is actually analogous to the role e^{At} plays in the solutions of the linear differential state equations that describe linear, continuous-time, time-invariant systems.

Notice that in (3), $k \geq 0$. This is so because A^k for $k < 0$ may not exist; this is the case, for example, when A is a singular matrix – it has at least one eigenvalue at the origin. In contrast, e^{At} exists for any t positive or negative. The implication is that in discrete-time systems we may not be able to determine uniquely the initial past state $x(0)$ from a current state value $x(k)$; in contrast, in continuous-time systems, it is always possible to go backwards in time.

There are several methods to calculate A^k that mirror the methods to calculate e^{At}. One could, for example, use similarity transformations, or the z-transform. When all eigenvectors of A are linearly independent (this is the case, e.g., when all eigenvalues λ_i of A are distinct), then a similarity transformation exists so that

$$PAP^{-1} = \tilde{A} = \text{diag}[\lambda_i].$$

Then

$$A^k = P^{-1}\tilde{A}^k P = P^{-1} \begin{bmatrix} \lambda_1^k & & \\ & \ddots & \\ & & \lambda_n^k \end{bmatrix} P.$$

Alternatively, using the z-transforms, $A^k = \mathcal{Z}^{-1}\{z(zI - A)^{-1}\}$. Also when the eigenvalues λ_i of A are distinct, then

$$A^k = \sum_{i=0}^{n} A_i \lambda_i^k,$$

where $A_i = v_i \tilde{v}_i$ with $v_i, \tilde{v}_i$ the right and left eigenvectors of A that correspond to λ_i. Note that

$$\begin{bmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_n \end{bmatrix} = \begin{bmatrix} v_1 \cdots v_n \end{bmatrix}^{-1},$$

$A_i \lambda_i^k$ are the modes of the system. One could also use the Cayley-Hamilton theorem to determine A^k.

System Response

Consider the description (1). The response can be easily derived by writing the equation for $k = 0, 1, 2, \ldots$ and substituting or formally by induction. It is

$$x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-(j+1)} Bu(j), \quad k > 0 \tag{6}$$

and

$$y(k) = C A^k x(0) + \sum_{j=0}^{k-1} C A^{k-(j+1)} Bu(j)$$
$$+ Du(k), \quad k > 0$$
$$y(0) = Cx(0) + Du(0). \tag{7}$$

Note that (6) can also be written as

$$x(k) = A^k x(0) + [B, AB, \cdots, A^{k-1}B] \begin{bmatrix} u(k-1) \\ \vdots \\ u(0) \end{bmatrix}. \tag{8}$$

Clearly the response is the sum of two components, one due to the initial condition (state response) and one due to the input (input response). This illustrates the linear system principle of superposition.

If the initial time is k_0 and (4) is used, then

$$y(k) = C A^{k-k_0} x(k_0) + \sum_{j=k_0}^{k-1} C A^{k-(j+1)} Bu(j)$$
$$+ Du(k), k > k_0$$
$$y(k_0) = Cx(k_0) + Du(k_0). \tag{9}$$

Equivalence of State Variable Descriptions

Given description (1), consider the new state vector $\tilde{x}$ where

$$\tilde{x}(k) = P x(k)$$

with $P \in \mathbb{R}^{n \times n}$ a real nonsingular matrix.

Substituting $x = P^{-1}\tilde{x}$ in (1), we obtain

$$\tilde{x}(k+1) = \tilde{A}\tilde{x}(k) + \tilde{B}u(k)$$
$$y(k) = \tilde{C}\tilde{x}(k) + \tilde{D}u(k) \tag{10}$$

where

$$\tilde{A} = PAP^{-1}, \ \tilde{B} = PB, \ \tilde{C} = CP^{-1}, \ \tilde{D} = D$$

The state variable descriptions (1) and (9) are called equivalent and P is the equivalence transformation matrix. This transformation corresponds to a change in the basis of the state space, which is a vector space. Appropriately selecting P one can simplify the structure of $\tilde{A}$ (= PAP^{-1}). It can be easily shown that equivalent description gives rise to the same discrete impulse response and transfer function.

Summary

State variable descriptions for discrete-time, time-invariant systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

Recommended Reading

The state variable descriptions received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman. For historical comments and extensive references, see Kailath (1980). The use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Franklin GF, Powell DJ, Workman ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley, Longman, Inc., Menlo Park, CA

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions

Panos J. Antsaklis
Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Discrete-time processes that can be modeled by linear difference equations with time-varying coefficients can be written in terms of state variable descriptions of the form $x(k+1) = A(k)x(k) + B(k)u(k)$, $y(k) = C(k)x(k) + D(k)u(k)$. The response of such systems due to a given input and initial conditions is derived. Equivalence of state variable descriptions is also discussed.

Keywords

Discrete-time; Linear systems; State variable descriptions; Time-varying

Introduction

Discrete-time systems arise in a variety of ways in the modeling process. There are systems that are inherently defined only at discrete points in time; examples include digital devices, inventory systems, and economic systems such as banking where interest is calculated and added to savings accounts at discrete time interval. There are also systems that describe continuous-time systems at discrete points in time; examples include simulations of continuous processes using digital computers and feedback control systems that employ digital controllers and give rise to sampled-data systems.

Dynamical processes that can be described or approximated by linear difference equations with time-varying coefficients can also be described, via a change of variables, by state variable descriptions of the form

$$x(k+1) = A(k)x(k) + B(k)u(k); \ x(k_0) = x_0$$
$$y(k) = C(k)x(k) + D(k)u(k).$$

(1)

Above, the state vector $x(k)$ ($k \in \mathbb{Z}$, the set of integers) is a column vector of dimension n ($x(k) \in \mathbb{R}^n$); the output is $y(k) \in \mathbb{R}^m$ and the input is $u(k) \in \mathbb{R}^m$. $A(k)$, $B(k)$, $C(k)$, and $D(k)$ are matrices with entries functions of time k, $A(k) = [a_{ij}(k)]$, $a_{ij}(k) : \mathbb{Z} \to \mathbb{R}$ ($A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times m}$, $C(k) \in \mathbb{R}^{p \times n}$, $D(k) \in \mathbb{R}^{p \times m}$). The vector difference equation in (1) is the state equation, while the algebraic equation is the output equation. Note that in the time-invariant case, $A(k) = A$, $B(k) = B$, $C(k) = C$, and $D(k) = D$.

The advantage of the state variable description (1) is that given an input $u(k)$, $k \geq k_0$ and an initial condition $x(k_0) = x_0$, the state trajectories or motions for $k \geq k_0$ can be conveniently characterized. To determine the expressions, we first consider the homogeneous state equation and the corresponding initial value problem.

Solving $x(k+1) = A(k)x(k)$; $x(k_0) = x_0$

Consider the homogenous equation

L

$$x(k + 1) = A(k)x(k); \quad x(k_0) = x_0 \quad (2)$$

Note that

$$x(k_0 + 1) = A(k_0)x(k_0)$$

$$x(k_0 + 2) = A(k_0 + 1)A(k_0)x(k_0)$$

$$\vdots$$

$$x(k) = A(k - 1)A(k - 2)\cdots A(k_0)x(k_0)$$

$$= \prod_{j=k_0}^{k-1} A(j)x(k_0), \quad k > k_0$$

This result can be shown formally by induction. The solution of (2) is then

$$x(k) = \Phi(k, k_0)x(k_0), \quad (3)$$

where $\Phi(k, k_0)$ is the *state transition matrix* of (2) given by

$$\Phi(k, k_0) = \prod_{j=k_0}^{k-1} A(j), \quad k > k_0; \ \ \Phi(k_0, k_0) = I.$$

$$(4)$$

Note that in the time-invariant case, $\Phi(k, k_0) = A^{k-k_0}$.

System Response

Consider now the state equation in (1). It can be easily shown that the solution is

$$x(k) = \Phi(k, k_0)x(k_0)$$

$$+ \sum_{j=k_0}^{k-1} \Phi(k, j + 1)B(j)u(j), \quad k > k_0,$$

$$(5)$$

and the response $y(k)$ of (1) is

$$y(k) = C(k)\Phi(k, k_0)x(k_0)$$

$$+ C(k) \sum_{j=k_0}^{k-1} \Phi(k, j + 1)B(j)u(j)$$

$$+ D(k)u(k), \quad k > k_0, \quad (6)$$

and

$$y(k_0) = C(k_0)x(k_0) + D(k_0)u(k_0).$$

Equation (5) is the sum of two parts, the *state response* (when $u(k) = 0$ and the system is driven only by the initial state conditions) and the *input response* (when $x(k_0) = 0$ and the system is driven only by the input $u(k)$); this illustrates the linear systems principle of superposition.

Equivalence of State Variable Descriptions

Given (1), consider the new state vector $\tilde{x}$ where

$$\tilde{x}(k) = P(k)x(k)$$

where $P^{-1}(k)$ exists. Then

$$\tilde{x}(k + 1) = \tilde{A}(k)\tilde{x}(k) + \tilde{B}(k)u(k)$$

$$y(k) = \tilde{C}(k)\tilde{x}(k) + \tilde{D}(k)u(k)$$

$$(7)$$

where

$$\tilde{A}(k) = P(k + 1)A(k)P^{-1}(k),$$

$$\tilde{B}(k) = P(k + 1)B(k),$$

$$\tilde{C}(k) = C(k)P^{-1}(k),$$

$$\tilde{D}(k) = D(k)$$

is equivalent to (1). It can be easily shown that equivalent descriptions give rise to the same discrete impulse responses.

Summary

State variable descriptions for linear discrete-time time-varying systems were introduced and the state and output responses to inputs and initial conditions were derived. The equivalence of state variable representations was also discussed.

Cross-References

▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Recommended Reading

The state variable descriptions received wide acceptance in systems theory beginning in the late 1950s. This was primarily due to the work of R.E. Kalman. For historical comments and extensive references, see Kailath (1980). The use of state variable descriptions in systems and control opened the way for the systematic study of systems with multi-inputs and multi-outputs.

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Astrom KJ, Wittenmark B (1997) Computer controlled systems: Theory and Design, 3rd edn. Prentice Hall, Upper Saddle River, NJ

Franklin GF, Powell DJ, Workman ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley, Menlo Park, CA

Jury EI (1958) Sampled-data control systems. Wiley, New York

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

LMI Approach to Robust Control

Kang-Zhi Liu
Department of Electrical and Electronic Engineering, Chiba University, Chiba, Japan

Abstract

In the analysis and design of robust control systems, LMI method plays a fundamental role. This article gives a brief introduction to this topic. After the introduction of LMI, it is illustrated how a control design problem is related with matrix inequality. Then, two methods are explained on how to transform a control problem characterized by matrix inequalities to LMIs, which is the core of the LMI approach. Based on this knowledge, the LMI solutions to various kinds of robust control problems are illustrated. Included are $\mathcal{H}_\infty$ and $\mathcal{H}_2$ control, regional pole placement, and gain-scheduled control.

Keywords

Introduction of LMI

A matrix inequality in a form of

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0 \qquad (1)$$

is called an LMI (linear matrix inequality). Here, $x = [x_1 \cdots x_m]$ is the unknown vector and $F_i (i = 1, \ldots, m)$ is a symmetric matrix. $F(x)$ is an affine function of x. The inequality means that $F(x)$ is positive definite.

LMI can be solved effectively by numerical algorithms such as the famous interior point method (Nesterov and Nemirovskii 1994). MATLAB has an LMI toolbox (Gahinet et al. 1995) tailored for solving the related control problems. Boyd et al. (1994) provide detailed theoretic fundamentals of LMI. A comprehensive and up-to-date treatment on the applications of LMI in robust control is covered in Liu and Yao (2014).

The notation $\text{He}(A) = A + A^T$ is used to simplify the presentation of large matrices; $A_\perp$ is a matrix whose columns form the basis of the kernel space of A, i.e., $AA_\perp = 0$. Further, $A \otimes B$ denotes the Kronecker product of matrices (A, B).

L

Control Problems and LMI

In control problems, it is often the case that the variables are matrices. For example, the necessary and sufficient condition for the stability of a linear system $\dot{x}(t) = Ax(t)$ is that there exists a positive-definite matrix P satisfying the inequality $AP + PA^T < 0$. Although this is different from the LMI of Eq. (1) in form, it can be converted to Eq. (1) equivalently by using a basis of symmetric matrices.

Next, consider the stabilization of system $\dot{x} = Ax + Bu$ by a state feedback $u = Fx$. The closed-loop system is $\dot{x} = (A + BF)x$. Therefore, the stability condition is that there exist a positive-definite matrix P and a feedback gain matrix F satisfying the inequality

$$(A + BF)P + P(A + BF)^T < 0. \quad (2)$$

In this inequality, FP, the product of unknown variables F and P, appears. Such matrix inequality is called a *bilinear matrix inequality*, or *BMI* for short. BMI problem is non-convex and difficult to solve. There are mainly two methods for transforming a BMI into an LMI: variable elimination and variable change.

From BMI to LMI: Variable Elimination

The method of variable elimination is good at optimizing single-objective problems. This method is based on the theorem below (Gahinet and Apkarian 1994).

Lemma 1 *Given real matrices E, F, G with G being symmetric, the inequality*

$$E^T X F + F^T X^T E + G < 0 \quad (3)$$

has a solution X if and only if the following two inequalities hold simultaneously

$$E_\perp^T G E_\perp < 0, \quad F_\perp^T G F_\perp < 0. \quad (4)$$

Application of this theorem to the previous stabilization problem (2) yields $(B^T)_\perp^T (AP + PA^T)(B^T)_\perp < 0$, which is an LMI about P. Once P is obtained, it may be substituted back into the inequality (2) and solve for F.

For output feedback problems, it is often needed to construct a new matrix from two given matrices in solving a control problem with LMI approach. The method is given by the following lemma.

Lemma 2 *Given two n-dimensional positive-definite matrices X and Y, a 2n-dimensional positive-definite matrix $\mathbb{P}$ satisfying the conditions*

$$\mathbb{P} = \begin{bmatrix} Y & * \\ * & * \end{bmatrix}, \quad \mathbb{P}^{-1} = \begin{bmatrix} X & * \\ * & * \end{bmatrix}$$

can be constructed if and only if

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0. \quad (5)$$

Factorizing $Y - X^{-1}$ as FF^T, a solution is given by

$$\mathbb{P} = \begin{bmatrix} Y & F \\ F^T & I \end{bmatrix}.$$

As an example of output feedback control design, let us consider the stabilization of the plant

$$\dot{x}_P = Ax_P + Bu, \ y = Cx_P \quad (6)$$

with a full-order dynamic controller

$$\dot{x}_K = A_K x_K + B_K y, \ u = C_K x_K + D_K y. \quad (7)$$

The closed-loop system is

$$\begin{bmatrix} \dot{x}_P \\ \dot{x}_K \end{bmatrix} = A_c \begin{bmatrix} x_P \\ x_K \end{bmatrix}, A_c = \begin{bmatrix} A + BD_K C & BC_K \\ B_K C & A_K \end{bmatrix}. \quad (8)$$

The stability condition is that the matrix inequality

$$A_c^T \mathbb{P} + \mathbb{P} A_c < 0 \quad (9)$$

has a solution $\mathbb{P} > 0$. To apply the variable elimination method, we need to put all coefficient matrices of the controller into in a single matrix. This is done as follows:

$$A_c = \overline{A} + \overline{B}\mathcal{K}\overline{C}, \ \mathcal{K} = \begin{bmatrix} D_K & C_K \\ B_K & A_K \end{bmatrix} \quad (10)$$

in which $\overline{A} = \text{diag}(A, 0)$, $\overline{B} = \text{diag}(B, I)$, and $\overline{C} = \text{diag}(C, I)$, all being block diagonal. Then, based on Lemma 1, the stability condition reduces to the existence of symmetric matrices X, Y satisfying LMIs

$$(B^T)_{\perp}^T (AX + XA^T)(B^T)_{\perp} < 0 \quad (11)$$

$$(C_{\perp})^T (YA + A^T Y)C_{\perp} < 0. \quad (12)$$

Meanwhile, the positive definiteness of matrix $\mathbb{P}$ is guaranteed by Eq. (5) in Lemma 2.

From BMI to LMI: Variable Change

We may also use the method of variable change to transform a BMI into an LMI. This method is good at multi-objective optimization.

The detail is as follows (Gahinet 1996). A positive-definite matrix can always be factorized as the quotient of two triangular matrices, i.e.,

$$\mathbb{P}\Pi_1 = \Pi_2, \Pi_1 = \begin{bmatrix} X & I \\ M^T & 0 \end{bmatrix}, \Pi_2 = \begin{bmatrix} I & Y \\ 0 & N^T \end{bmatrix}. \quad (13)$$

$\mathbb{P} > 0$ is guaranteed by Eq. (5) for a full-order controller. Further, the matrices M, N are computed from $MN^T = I - XY$. Consequently, they are nonsingular.
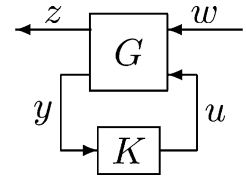
An equivalent inequality $\Pi_1^T A_c^T \Pi_2 + \Pi_2^T A_c \Pi_1 < 0$ is obtained by multiplying Eq. (9) with Π_1^T and Π_1. After a change of variables, this inequality reduces to an LMI

$$\text{He} \begin{bmatrix} AX + B\mathbb{C} & A + B\mathbb{D}C + \mathbb{A}^T \\ 0 & YA + \mathbb{B}C \end{bmatrix} < 0. \quad (14)$$

The new variables $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}$ are set as

LMI Approach to Robust Control, Fig. 1 Generalized feedback system

$$\begin{aligned} \mathbb{A} &= NA_K M^T + NB_K CX + YBC_K M^T \\ &\quad + Y(A + BD_K C)X \\ \mathbb{B} &= NB_K + YBD_K, \quad \mathbb{C} = C_K M^T \\ &\quad + D_K CX, \quad \mathbb{D} = D_K. \end{aligned} \quad (15)$$

The coefficient matrices of the controller become

$$\begin{aligned} D_K &= \mathbb{D}, C_K = (\mathbb{C} - D_K CX)M^{-T}, \\ B_K &= N^{-1}(\mathbb{B} - YBD_K) \\ A_K &= N^{-1}(\mathbb{A} - NB_K CX - YBC_K M^T \\ &\quad - Y(A + BD_K C)X)M^{-T}. \end{aligned} \quad (16)$$

$\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control

In system optimization, $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms are the most popular and effective performance indices. $\mathcal{H}_2$ norm of a transfer function is closely related with the squared area of its impulse response. So, a smaller $\mathcal{H}_2$ norm implies a faster response. Meanwhile, $\mathcal{H}_\infty$ norm of a transfer function is the largest magnitude of its frequency response. Hence, for a transfer function from the disturbance to the controlled output, a smaller $\mathcal{H}_\infty$ norm guarantees a better disturbance attenuation.

Usually $\mathcal{H}_2$ and $\mathcal{H}_\infty$ optimization problems are treated in the generalized feedback system of Fig. 1. Here, the generalized plant $G(s)$ includes the nominal plant, the performance index, and the weighting functions.

Let the generalized plant $G(s)$ be

$$G(s) = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} (sI - A)^{-1} \begin{bmatrix} B_1 & B_2 \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{bmatrix}. \quad (17)$$

Further, the stabilizability of (A, B_2) and the detectability of (C_2, A) are assumed. The

closed-loop transfer matrix from the disturbance w to the performance output z is denoted by

$$H_{zw}(s) = C_c(sI - A_c)^{-1}B_c + D_c. \qquad (18)$$

The condition for $H_{zw}(s)$ to have an $\mathcal{H}_2$ norm less than γ, i.e., $\|H_{zw}\|_2 < \gamma$, is that there are symmetric matrices $\mathbb{P}$ and W satisfying

$$\begin{bmatrix} \mathbb{P}A_c + A_c^T\mathbb{P} & C_c^T \\ C_c & -I \end{bmatrix} < 0, \quad \begin{bmatrix} W & B_c^T\mathbb{P} \\ \mathbb{P}B_c & \mathbb{P} \end{bmatrix} > 0 \qquad (19)$$

as well as $\mathrm{Tr}(W) < \gamma^2$. Here, $\mathrm{Tr}(W)$ denotes the trace of matrix W, i.e., the sum of its diagonal entries.

The LMI solution is derived via the application of the variable change method, as given below.

Theorem 1 *Suppose that $D_{11} = 0$. The $\mathcal{H}_2$ control problem is solvable if and only if there exist symmetric matrices X, Y, W and matrices $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$ satisfying the following LMIs:*

$$\mathrm{He}\begin{bmatrix} AX + B_2\mathbb{C} & 0 & 0 \\ A^T + \mathbb{A} & YA + \mathbb{B}C_2 & 0 \\ C_1X + D_{12}\mathbb{C} & C_1 & -\frac{1}{2}I \end{bmatrix} < 0 \qquad (20)$$

$$\begin{bmatrix} W & B_1^T & B_1^TY \\ B_1 & X & I \\ YB_1 & I & Y \end{bmatrix} > 0, \ \mathrm{Tr}(W) < \gamma^2. \qquad (21)$$

When the LMI Eqs. (20) *and* (21) *have solutions, an $\mathcal{H}_2$ controller is given by* Eq. (16) *by setting* $\mathbb{D} = 0$.

The $\mathcal{H}_\infty$ control problem is to design a controller so that $\|H_{zw}\|_\infty < \gamma$. The starting point of $\mathcal{H}_\infty$ control is the famous bounded real lemma, which states that $H_{zw}(s)$ has an $\mathcal{H}_\infty$ norm less than γ if and only if there is a positive-definite matrix $\mathbb{P}$ satisfying

$$\begin{bmatrix} A_c^T\mathbb{P} + \mathbb{P}A_c & \mathbb{P}B_c & C_c^T \\ B_c^T\mathbb{P} & -\gamma I & D_c^T \\ C_c & D_c & -\gamma I \end{bmatrix} < 0. \qquad (22)$$

There are two kinds of LMI solutions to this control problem: one based on variable elimination and one based on variable change.

To state the first solution, define the following matrices first:

$$N_Y = [C_2 \ D_{21}]_\perp, \ \ N_X = [B_2^T \ D_{12}^T]_\perp. \qquad (23)$$

Theorem 2 *The $\mathcal{H}_\infty$ control problem has a solution if and only if* Eq. (5) *and the following LMIs have positive-definite solutions X, Y:*

$$\begin{bmatrix} N_X^T & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} AX + XA^T & XC_1^T & B_1 \\ C_1X & -\gamma I & D_{11} \\ B_1^T & D_{11}^T & -\gamma I \end{bmatrix}$$
$$\times \begin{bmatrix} N_X & 0 \\ 0 & I \end{bmatrix} < 0 \qquad (24)$$

$$\begin{bmatrix} N_Y^T & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} YA + A^TY & YB_1 & C_1^T \\ B_1^TY & -\gamma I & D_{11}^T \\ C_1 & D_{11} & -\gamma I \end{bmatrix}$$
$$\times \begin{bmatrix} N_Y & 0 \\ 0 & I \end{bmatrix} < 0. \qquad (25)$$

Once a matrix $\mathbb{P}$ is computed according to Lemma 2, Eq. (22) becomes an LMI and its solution yields the controller.

The second solution is given below.

Theorem 3 *The $\mathcal{H}_\infty$ control problem has a solution if and only if* Eq. (5) *and the following LMI have solutions X, Y and $\mathbb{A}$, $\mathbb{B}$, $\mathbb{C}$, $\mathbb{D}$:*

$$\mathrm{He}\begin{bmatrix} AX + B_2\mathbb{C} & A + B_2\mathbb{D}C_2 & B_1 + B_2\mathbb{D}D_{21} & 0 \\ \mathbb{A} & YA + \mathbb{B}C_2 & YB_1 + \mathbb{B}D_{21} & 0 \\ 0 & 0 & -\frac{\gamma}{2}I & 0 \\ C_1X + D_{12}\mathbb{C} & C_1 + D_{12}\mathbb{D}C_2 & D_{11} + D_{12}\mathbb{D}D_{21} & -\frac{\gamma}{2}I \end{bmatrix} < 0. \qquad (26)$$

The controller is given by Eq. (16).

Regional Pole Placement

The location of system poles determines the response quality. However, for uncertain systems it is impossible to place the closed-loop poles at fixed points because they move with the variation of the plant. Nevertheless, it is still possible to place the closed-loop poles inside a region. For convex regions characterized by LMI, the design method is mature and proven effective in practice.

Let us see how to characterize a convex region. It is easy to know that a complex number z is inside the disk of Fig. 2a if and only if it satisfies

$$\begin{bmatrix} -r & z+c \\ \bar{z}+c & -r \end{bmatrix} < 0.$$

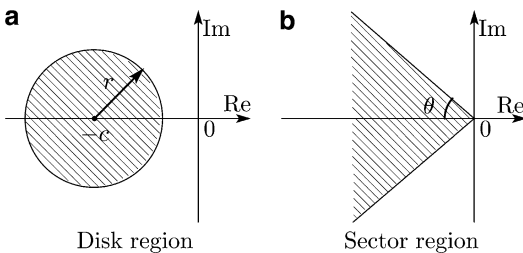Similarly, z is inside the sector of Fig. 2b if and only if

$$\begin{bmatrix} (z+\bar{z})\sin\theta & (z-\bar{z})\cos\theta \\ -(z-\bar{z})\cos\theta & (z+\bar{z})\sin\theta \end{bmatrix} < 0.$$

Generally, the set of complex number z characterized by

$$D = \{z \in \mathbb{C} | L + zM + \bar{z}M^T < 0\} \qquad (27)$$

is called an LMI region, in which L is a symmetric matrix. For the dynamic system

$$\dot{x} = Ax, \qquad (28)$$

all of its poles are in the LMI region D if and only if there is a positive-definite matrix $\mathbb{P}$ satisfying the LMI

$$L \otimes \mathbb{P} + M \otimes (A\mathbb{P}) + M^T \otimes (A\mathbb{P})^T < 0. \quad (29)$$

This forms the basis for the regional pole placement design.

For the disk region in Fig. 2a, the condition becomes

$$\begin{bmatrix} -r\mathbb{P} & c\mathbb{P}+A\mathbb{P} \\ c\mathbb{P}+(A\mathbb{P})^T & -r\mathbb{P} \end{bmatrix} < 0. \qquad (30)$$

Meanwhile, for the sector region in Fig. 2b, the corresponding LMI is

$$\begin{bmatrix} (A\mathbb{P}+\mathbb{P}A^T)\sin\theta & (A\mathbb{P}-\mathbb{P}A^T)\cos\theta \\ -(A\mathbb{P}-\mathbb{P}A^T)\cos\theta & (A\mathbb{P}+\mathbb{P}A^T)\sin\theta \end{bmatrix} < 0. \tag{31}$$

Moreover, for a composite LMI region, such as the intersection of the disk and the sector, the pole placement is guaranteed by enforcing a common solution $\mathbb{P}$ to all the corresponding LMIs.

In the pole placement design, only the variable change method is applicable. For example, in the nominal closed-loop system Eq. (8), the pole placement condition is that the LMI

$$L \otimes \begin{bmatrix} X & I \\ I & Y \end{bmatrix}$$
$$+ \text{He}\left(M \otimes \begin{bmatrix} AX+B\mathbb{C} & A+B\mathbb{D}C \\ \mathbb{A} & YA+\mathbb{B}C \end{bmatrix} \right) < 0 \tag{32}$$

and Eq. (5) are solvable (Chilali and Gahinet 1996).

For systems with norm-bounded parameter uncertainty, a robust pole placement method is provided in Chilali et al. (1999).

Multi-objective Control

It is noted that all of the preceding control designs involve a positive-definite matrix $\mathbb{P}$. Therefore, a multi-objective control design is easily realized

LMI Approach to Robust Control, Fig. 2 Typical examples of LMI region

L

by enforcing a common solution $\mathbb{P}$ to the corresponding matrix inequality conditions.

Gain-Scheduled Control

In practice, many nonlinear systems can be expressed as linear systems with state-dependent coefficients in form, which is known as the LPV (linear parameter-varying) form. For example, in the model of a robot arm $J\ddot{\theta} + mgl\sin\theta = u$, if we define a parameter as $p(t) = \sin\theta/\theta$, then it can be written as an LPV model $J\ddot{\theta}(t) + mglp(t)\theta(t) = u(t)$. In this class of systems, when the parameter $p(t)$ is available online and its range is finite, one may tune the controller parameters based on the information of $p(t)$, so as to achieve a higher performance. This is referred to as gain-scheduled control.

Consider the following affine model:

$$\dot{x} = A(p(t))x + B_1(p(t))d + B_2(p(t))u \quad (33)$$

$$z = C_1(p(t))x + D_{11}d + D_{12}u \quad (34)$$

$$y = C_2(p(t))x + D_{21}d \quad (35)$$

where $A(p) \sim C_2(p)$ are affine functions of the time-varying parameter vector $p(t)$, such as $A(p) = A_0 + \sum_{i=1}^{q} p_i(t)A_i$. The gain-scheduled control is to impose, on the coefficient matrices of the controller, the same affine structure about $p(t)$ such as $A_K(p) = A_{K0} + \sum_{i=1}^{q} p_i(t)A_{Ki}$.

To simplify the design, it is desirable that the coefficient matrices of the closed-loop system become affine functions of the parameter vector $p(t)$. This may be satisfied by restraining some of the matrices of the controller to constant ones. The easy-to-design structure of a gain-scheduled controller is summarized as follows:

- Both $B_2(p)$ and $C_2(p)$ depend on $p(t)$: (B_K, C_K) must be constant matrices besides $D_K = 0$.
- Constant (B_2, C_2): All coefficient matrices of the controller can be affine functions of the parameter vector $p(t)$.
- Constant B_2: (B_K, D_K) must be constant matrices.
- Constant C_2: (C_K, D_K) must be constant matrices.

When the structure of the gain-scheduled controller is chosen as summarized above, the solvability conditions reduce to those at all vertices θ_i of the scheduling parameter vector $p(t)$. Further, a multi-objective is achieved by imposing a common solution $\mathbb{P}$ to all LMI conditions. Some concrete examples are illustrated below:

$\mathcal{H}_\infty$ *Norm Spec*: The conditions of Theorem 3 are satisfied at all vertices θ_j of the parameter vector $p(t)$.

$\mathcal{H}_2$ *Norm Spec*: The conditions of Theorem 1 are satisfied at all vertices θ_j of the parameter vector $p(t)$.

Regional Pole Placement: Eq. (32) is satisfied at all vertices θ_j of the parameter vector $p(t)$ and Eq. (5) holds.

Moreover, a different gain-scheduled method is proposed in Packard (1994) for parametric systems with norm-bounded uncertainty.

Summary and Future Direction

LMI approach is a very powerful method that can be applied to solve most of the robust control problems smartly and effectively. In particular, its capability of handling the multi-objective control problems is very attractive and proven useful in industrial applications.

Further study is needed in the following directions.

- New method of variable change is desired in order to deal with the robust performance design of parametric systems.
- Almost all robust performance designs are carried out based on sufficient conditions. It is very important to discover less conservative design methods.

Cross-References

▶ H-Infinity Control
▶ Linear Matrix Inequality Techniques in Optimal Control
▶ Optimization Based Robust Control
▶ Optimization-Based Control Design Techniques and Tools
▶ Robust Synthesis and Robustness Analysis Techniques and Tools

Bibliography

Apkarian P, Gahinet P (1995) A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. IEEE Trans Autom Control 40(5):853–864

Boyd SP et al (1994) Linear matrix inequalities in system and control. SIAM, Philadelphia

Chilali M, Gahinet P (1996) $\mathcal{H}_\infty$ design with pole placement constraints: an LMI approach. IEEE Trans Autom Control 41(3):358–367

Chilali M, Gahinet P, Apkarian P (1999) Robust pole placement in LMI regions. IEEE Trans Autom Control 44(12):2257–2270

Gahinet P (1996) Explicit controller formulas for LMI-based $\mathcal{H}_\infty$ synthesis. Automatica 32(7):1007–1014

Gahinet P, Apkarian P (1994) A linear matrix inequality approach to $\mathcal{H}_\infty$ control. Int J Robust Nonlinear Control 4:421–448

Gahinet P, Nemirovski A, Laub AJ, Chilali M (1995) LMI control toolbox. The MathWorks, Inc., Natick

Liu KZ, Yao Y (2014, to appear) Robust control: theory and applications. Wiley, New York

Nesterov Y, Nemirovskii A (1994) Interior-point polynomial methods in convex programming. SIAM, Philadelphia

Packard A (1994) Gain scheduling via linear fractional transformations. Syst Control Lett 22: 79–92

LTI Systems

▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions

Lyapunov Methods in Power System Stability

Hsiao-Dong Chiang
School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA

Abstract

Energy functions, an extension of Lyapunov functions, have been used in electric power systems for several applications. An overview of energy function theory for general nonlinear autonomous dynamical systems along with its applications to electric power systems is presented. The issue of how to optimally determine the critical level value of an energy function for estimating stability regions of nonlinear dynamical systems is also addressed.

Keywords

Introduction

Energy functions, an extension of the Lyapunov functions, have been practically used in electric power systems for several applications. A comprehensive energy function theory for general nonlinear autonomous dynamical systems along with its applications to electric power systems will be summarized in this article.

We consider a general nonlinear autonomous dynamical system described by the following equation:

$$\dot{x}(t) = f(x(t)) \tag{1}$$

We say a function $V : R^n \to R$ is an energy function for the system (1) if the following three conditions are satisfied (Chiang et al. 1987):

(E1): The derivative of the energy function $V(x)$ along any system trajectory $x(t)$ is nonpositive, i.e., $\dot{V}(x(t)) \leq 0$.

(E2): If $x(t)$ is a nontrivial trajectory (i.e., $x(t)$ is not an equilibrium point), then along the nontrivial trajectory $x(t)$ the set $\{t \in R : \dot{V}(x(t)) = 0\}$ has measure zero in R.

(E3): That a trajectory $x(t)$ has a bounded value of $V(x(t))$ for $t \in R^+$ implies that the trajectory $x(t)$ is also bounded.

Condition (E1) indicates that the value of an energy function is nonincreasing along its trajectory, but does not imply that the energy function is strictly decreasing along any trajectory. Conditions (E1) and (E2) imply that the energy function is strictly decreasing along any system trajectory. Property (E3) states that the energy function is a proper map along any system trajectory but need not be a proper map for the entire state space. Obviously, an energy function may not be a Lyapunov function.

As an illustration of the energy function, we consider the following classical transient stability model and derive an energy function for the model. Consider a power system consisting of n generators. Let the loads be modeled as constant impedances. Under the assumption that the transfer conductance of the reduced network after eliminating all load buses is zero, the dynamics of the ith generator can be represented by the equations

$$\dot{\delta}_i = \omega_i$$

$$M_i \dot{\omega}_i = P_i - D_i \omega_i - \sum_{j=1} V_i V_j B_{ij} \sin(\delta_i - \delta_j)$$

$$(2)$$

where the voltage at node $i+1$ is served as the reference, i.e., $\delta_{i+1} := 0$. This is a version of the so-called *classical model* of the power system. It can be shown that the following function is an energy function $V(\delta, \omega)$ which satisfies conditions (E1)–(E3) for the classical model (2).

$$V(\delta, \omega) = \frac{1}{2} \sum_{i=1}^{n} M_i \omega_i^{\mathbf{2}} - \sum_{i=1}^{n} P_i \left(\delta_i - \delta_j^s \right)$$
$$- \sum_{i=1}^{n} \sum_{j=i+1}^{n+1} V_i V_j B_{ij} \mathbf{cos}(\delta_i - \delta_j)$$
$$- \mathbf{cos}(\delta_i^s - \delta_j^s) \qquad (3)$$

where $x^s = (\delta^s, \mathbf{0})$ is the stable equilibrium point under consideration.

Energy Function Theory

In general, the dynamical behaviors of trajectories of general nonlinear systems can be very complicated. The asymptotical behaviors (i.e., the ω-limit set) of trajectories can be quasiperiodic trajectories or chaotic trajectories. However, as shown below, every trajectory of system (1) having an energy function has only two modes of behaviors: its trajectory either converges to an equilibrium point or goes to infinity (becomes unbounded) as time increases. This result is explained in the following theorem:

Theorem 1 (Global Behavior of Trajectories) *If there exists a function satisfying condition (E1) and condition (E2) of the energy function for system (1), then every bounded trajectory of system (1) converges to one of the equilibrium points.*

Theorem 1 asserts that there does not exist any limit cycle (oscillatory behavior) or bounded complicated behavior such as almost periodic trajectory, chaotic motion, etc. in the system. We next show a sharper result, asserting that every trajectory on the stability boundary must converge to one of the unstable equilibrium points (UEPs) on the stability boundary. Recall that for a hyperbolic equilibrium point, it is an *(asymptotically) stable equilibrium point* if all the eigenvalues of its corresponding Jacobian have negative real parts; otherwise it is an *unstable equilibrium point*. Let $\hat{x}$ be a hyperbolic equilibrium point. Its stable and unstable manifolds, $W^s(\hat{x})$ and $W^u(\hat{x})$, are well defined. There are many physical systems such as electric power systems containing multiple stable equilibrium points. A useful concept for these kinds of systems is that of the *stability region* (also called the *region of attraction*). The stability region of a stable equilibrium point x_s is defined as

$$A(x_s) := \left\{ x \in R^n : \lim_{t \to \infty} \mathbf{\Phi}_t(x) = x_s \right\}$$

The boundary of stability region $A(x_s)$ is called the *stability boundary* of (x_s) and will be denoted by $\partial A(x_s)$.

Theorem 2 (Trajectories on the Stability Boundary (Chiang et al. 1987)) *If there exists an energy function for system (1), then every trajectory on the stability boundary $\partial A(x_s)$ converges to one of the equilibrium points on the stability boundary $\partial A(x_s)$.*

The significance of this theorem is that it offers an effective way to characterize the stability boundary. In fact, Theorem 2 asserts that the stability boundary $\partial A(x_s)$ is contained in the union of stable manifolds of the UEPs on the stability boundary, i.e.,

$$\partial A(x_s) \subseteq \bigcup_{x_i \in \{E \cap \partial A(x_s)\}} W^s(x_i)$$

The following two theorems give interesting results on the structure of the equilibrium points on the stability boundary. Moreover, it presents a necessary condition for the existence of certain types of equilibrium points on a *bounded* stability boundary.

Theorem 3 (Structure of Equilibrium Points on the Stability Boundary (Chiang and Thorp 1989)) *If there exists an energy function for system (1) which has an asymptotically stable equilibrium point x_s (but not globally asymptotically stable), then the stability boundary (x_s) must contain at least one type one equilibrium point. If, furthermore, the stability region is bounded, then the stability boundary $\partial A(x_s)$ must contain at least one type one equilibrium point and one source.*

Theorem 4 (Sufficient Condition for Unbounded Stability Region (Chiang et al. 1987)) *If there exists an energy function for system (1) which has an asymptotically stable equilibrium point x_s (but not globally asymptotically stable) and if $\partial A(x_s)$ contains no source, then the stability region $A(x_s)$ is unbounded.*

A direct application of this is that the stability boundary $\partial A(x_s)$ of an (asymptotically) stable equilibrium point of the classical power system stability model (2) is unbounded.

Optimally Estimating Stability Region Using Energy Functions

In this section, we focus on how to optimally determine the critical level value of an energy function for estimating the stability boundary $\partial A(x_s)$. We consider the following set:

$$S_v(k) = \{x \in R^n : V(x) < k\} \qquad (4)$$

where $V(.) : R^n \rightarrow R$ is an energy function. We shall call the boundary of set (2) $\partial S(k) := \{x \in R^n : V(x) = k\}$ the *level set* (or *constant energy surface*) and k the *level value*. Generally speaking, this set $S(k)$ can be very complicated with several connected components even for the 2-dimensional case. We use the notation $S_k(x_s)$ to denote the only component of the several disjoint connected components of S_k that contains the stable equilibrium point x_s.

Theorem 5 (Optimal Estimation) *Consider the nonlinear system (1) which has an energy function $V(x)$. Let x_s be an asymptotically stable equilibrium point whose stability region $A(x_s)$ is not dense in R^n. Let E_1 be the set of type one equilibrium points and $\hat{c} = \min_{x_i \in \partial A(x_s) \cap_1^E} V(x_i)$, and then*
1. *$S_{\hat{c}}(x_s) \subset A(x_s)$.*
2. *The set $\{S_b(x_s) \cap \bar{A}^c(x_s)\}$ is nonempty for any number $b > c$.*

This theorem leads to an optimal estimation of the stability region $A(x_s)$ via an energy function $V(.)$ (Chiang and Thorp 1989). For the purpose of illustration, we consider the following simple example:

$$\dot{x}_1 = -\sin x_i - 0.5 \sin(x_1 - x_2) + 0.01$$
$$\dot{x}_2 = -0.5 \sin x_2 - 0.5 \sin(x_2 - x_1) + 0.05 \quad (5)$$

It is easy to show that the following function is an energy function for system (5):

$$V(x_1, x_2) = -2\cos x_1 - \cos x_2 - \cos(x_1 - x_2)$$
$$-0.02x_1 - 0.1x_2 \qquad (6)$$

The point $x^s \left(x_1^s, x_2^s\right) = (0.02801, 0.06403)$ is the stable equilibrium point whose stability region is to be estimated. Applying the optimal scheme to system (5), we have the critical level value of -0.31329. The Curve A in Fig. 1 is the exact stability boundary $\partial A(x^s)$ while Curve B is the stability boundary estimated by the connected component (containing the s.e.p. x^s) of the constant energy surface. It can be seen that the critical level value, -0.31329, is indeed the optimal value.
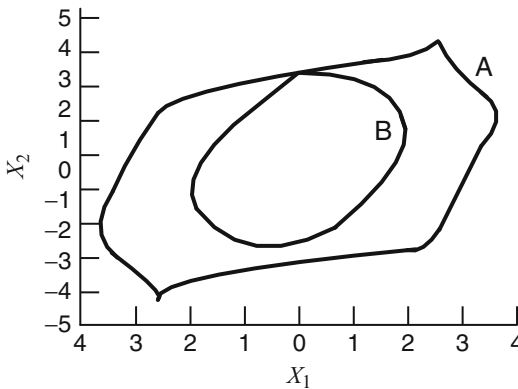
Lyapunov Methods in Power System Stability, Fig. 1
Curve A is the exact stability boundary $\partial A(x^s)$ of system
(5), while Curve B is the stability boundary estimated by
the constant energy surface (with level value of -0.31329)
of the energy function

Constructing Analytical Energy Functions for Transient Stability Models

The task of constructing an energy function for
a (post-fault) transient stability model is essential
to direct stability analysis of power systems. The
role of the energy function is to make feasible
a direct determination of whether a given point
(such as the initial point of a post-fault power sys-
tem) lies inside the stability region of post-fault
SEP without performing numerical integration. It
has been shown that a general (analytical) energy
function for power systems with losses does not
exist (Chiang 1989). One key implication is that
any general procedure attempting to construct
an energy function for a lossy power system
transient stability model must include a step that
checks for the existence of an energy function.
This step essentially plays the same role as the
Lyapunov equation in determining the stability of
an equilibrium point.

Several schemes are available for constructing
numerical energy functions for power system
transient stability models expressed as a set of
general differential-algebraic equations (DAEs)
(Chu and Chiang 1999, 2005).

Applications

After decades of research and development in
the energy-function-based direct methods and the
time-domain simulation approach, it has become
clear that the capabilities of direct methods and
that of the time-domain approach complement
each other. The current direction of development
is to include appropriate direct methods and time-
domain simulation programs within the body of
overall power system stability simulation pro-
grams (Chiang 1999, 2011; Chiang et al. 1995;
Fouad and Vittal 1991; Sauer and Pai 1998).
For example, the direct method provides the ad-
vantages of fast computational speed and energy
margins which make it a good complement to
the traditional time-domain approach. The en-
ergy margin and its functional relations to cer-
tain power system parameters are an effective
complement to develop tools such as preventive
control schemes for credible contingencies which
are unstable and to develop fast calculators for
available transfer capability limited by transient
stability.

An effective, theory-based methodology for
online screening and ranking of a large set
of contingencies at operating points obtained
from state estimators has been developed in
Chiang et al. (2013). A set of improved BCU
classifiers, along with their analytical basis,
has been developed. Extensive evaluation of the
improved BCU classifiers on a large test system
and on the actual PJM interconnection system
for a fast screening has been performed. This
evaluation study is the largest in terms of system
size, 14,500 buses and 3,000 generators, for a
practical online transient stability assessment
application. The evaluation results, performed
on a total number of 5.3 million contingencies,
were very promising in terms of speed, accuracy,
reliability, and robustness (Chiang et al. 2013).
This study also confirms the practicality of
theory-based methodology for online transient
stability assessment of large-scale power
systems; in particular, theory-based methods are
suitable for power system online applications
which demand speed, accuracy, reliability, and
robustness.

Cross-References

Recommended Reading

A recent book which contains a comprehensive treatment of energy functions theory and applications is Chiang (2011).

Bibliography

Chiang HD (1989) Study of the existence of energy functions for power systems with losses. IEEE Trans Circuits Syst CAS-36(11):1423–1429

Chiang HD (1999) Power system stability. In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering. Wiley, New York, pp 104–137

Chiang HD (2011) Direct methods for stability analysis of electric power systems: theoretical foundation, BCU methodologies, and applications. Wiley, Hoboken

Chiang HD, Thorp JS (1989) Stability regions of nonlinear dynamical systems: a constructive methodology. IEEE Trans Autom Control 34(12):1229–1241

Chiang HD, Wu FF, Varaiya PP (1987) Foundations of direct methods for power system transient stability analysis. IEEE Trans Circuits Syst CAS-34(2):160–173

Chiang HD, Chu CC, Cauley G (1995) Direct stability analysis of electric power systems using energy functions: theory, applications and perspective (invited paper)

Chiang HD, Li H, Tong J, Tada Y (2013) On-line transient stability screening of practical a 14,500-bus power system: methodology and evaluations. In: Khaitan SK, Gupta A (eds) High performance computing in power and energy systems. Springer, Berlin/New York, pp 335–358. ISBN:978-3-642-32682-0

Chu CC, Chiang HD (1999) Constructing analytical energy functions for network-reduction power systems with models: framework and developments. Circuits Syst Signal Process 18(1):1–16

Chu CC, Chiang HD (2005) Constructing analytical energy functions for network-preserving power system models. Circuits Syst Signal Process 24(4):363–383

Fouad AA, Vittal V (1991) Power system transient stability analysis: using the transient energy function method. Prentice-Hall, Englewood Cliffs

Sauer PW, Pai MA (1998) Power system dynamics and stability. Prentice-Hall, Upper Saddle River

Lyapunov's Stability Theory

Hassan K. Khalil
Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA

Abstract

Lyapunov's theory for characterizing and studying the stability of equilibrium points is presented for time-invariant and time-varying systems modeled by ordinary differential equations.

L

Keywords

Asymptotic stability; Equilibrium point; Exponential stability; Global asymptotic stability; Hurwitz matrix; Invariance principle; Linearization; Lipschitz condition; Lyapunov function; Lyapunov surface; Negative (semi-) definite function; Perturbed system; Positive (semi-) definite function; Region of attraction; Stability; Time-invariant system; Time-varying system

Introduction

Stability theory plays a central role in systems theory and engineering. For systems represented by state models, stability is characterized by studying the asymptotic behavior of the state variables near steady-state solutions, like equilibrium points or periodic orbits. In this article, Lyapunov's method for determining the stability of equilibrium points is introduced. The attractive features of the method include a solid

theoretical foundation, the ability to conclude stability without knowledge of the solution (no extensive simulation effort), and an analytical framework that makes it possible to study the effect of model perturbations and design feedback control. Its main drawback is the need to search for an auxiliary function that satisfies certain conditions.

Stability of Equilibrium Points

We consider a nonlinear system represented by the state model

$$\dot{x} = f(x) \tag{1}$$

where the n-dimensional **locally Lipschitz** function $f(x)$ is defined for all x in a domain $D \subset R^n$. A function $f(x)$ is locally Lipschitz at a point x_0 if it satisfies the **Lipschitz condition** $\|f(x) - f(y)\| \leq L\|x - y\|$ for all x, y in some neighborhood of x_0, where L is a positive constant and $\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$. The Lipschitz condition guarantees that Eq. (1) has a unique solution for given initial state $x(0)$. Suppose $\bar{x} \in D$ is an **equilibrium point** of Eq. (1); that is, $f(\bar{x}) = 0$. Whenever the state of the system starts at $\bar{x}$, it will remain at $\bar{x}$ for all future time. Our goal is to characterize and study the stability of $\bar{x}$. For convenience, we take $\bar{x} = 0$. There is no loss of generality in doing so because any equilibrium point $\bar{x}$ can be shifted to the origin via the change of variables $y = x - \bar{x}$. Therefore, we shall always assume that $f(0) = 0$ and study stability of the origin $x = 0$.

The equilibrium point $x = 0$ of Eq. (1) is **stable** if for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$ such that $\|x(0)\| < \delta$ implies that $\|x(t)\| < \varepsilon$, for all $t \geq 0$. It is **asymptotically stable** if it is stable and δ can be chosen such that $\|x(0)\| < \delta$ implies that $x(t)$ converges to the origin as t tends to infinity. When the origin is asymptotically stable, the **region of attraction** (also called region of asymptotic stability, domain of attraction, or basin) is defined as the set of all points x such that the solution of Eq. (1) that starts from x at

time $t = 0$ approaches the origin as t tends to ∞. When the region of attraction is the whole space, we say that the origin is **globally asymptotically stable**. A stronger form of asymptotic stability arises when there exist positive constants c, k, and λ such that the solutions of Eq. (1) satisfy the inequality

$$\|x(t)\| \leq k\|x(0)\|e^{-\lambda t}, \quad \forall \, t \geq 0 \tag{2}$$

for all $\|x(0)\| < c$. In this case, the equilibrium point $x = 0$ is said to be **exponentially stable**. It is said to be globally exponentially stable if the inequality is satisfied for any initial state $x(0)$.

Linear Systems
For the linear time-invariant system

$$\dot{x} = Ax \tag{3}$$

the stability properties of the origin can be determined by the location of the eigenvalues of A. The origin is stable if and only if all the eigenvalues of A satisfy $\text{Re}[\lambda_i] \leq 0$ and for every eigenvalue with $\text{Re}[\lambda_i] = 0$ and algebraic multiplicity $q_i \geq 2$, $\text{rank}(A - \lambda_i I) = n - q_i$, where n is the dimension of x and q_i is the multiplicity of λ_i as a zero of $\det(\lambda I - A)$. The origin is globally exponentially stable if and only if all eigenvalues of A have negative real parts; that is, A is a **Hurwitz matrix**. For linear systems, the notions of asymptotic and exponential stability are equivalent because the solution is formed of exponential modes. Moreover, due to linearity, if the origin is exponentially stable, then the inequality of Eq. (2) will hold for all initial states.

Linearization
Suppose the function $f(x)$ of Eq. (1) is continuously differentiable in a domain D containing the origin. The Jacobian matrix $[\partial f/\partial x]$ is an $n \times n$ matrix whose (i, j) element is $\partial f_i/\partial x_j$. Let A be the Jacobian matrix evaluated at the origin $x = 0$. It can be shown that

$$f(x) = [A + G(x)]x, \quad \text{where } \lim_{x \to 0} G(x) = 0$$

This suggests that in a small neighborhood of the origin we can approximate the nonlinear system $\dot{x} = f(x)$ by its linearization about the origin $\dot{x} = Ax$. Indeed, we can draw conclusions about the stability of the origin as an equilibrium point for the nonlinear system by examining the eigenvalues of A. The origin of Eq. (1) is exponentially stable if and only if A is Hurwitz. It is unstable if $\text{Re}[\lambda_i] > 0$ for one or more of the eigenvalues of A. If $\text{Re}[\lambda_i] \leq 0$ for all i, with $\text{Re}[\lambda_i] = 0$ for some i, we cannot draw a conclusion about the stability of the origin of Eq. (1).

Lyapunov's Method

Let $V(x)$ be a continuously differentiable scalar function defined in a domain $D \subset R^n$ that contains the origin. The function $V(x)$ is said to be **positive definite** if $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$. It is said to be **positive semidefinite** if $V(x) \geq 0$ for all x. A function $V(x)$ is said to be **negative definite** or **negative semidefinite** if $-V(x)$ is positive definite or positive semidefinite, respectively. The derivative of V along the trajectories of Eq. (1) is given by

$$\dot{V}(x) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \dot{x}_i = \frac{\partial V}{\partial x} f(x)$$

where $[\partial V/\partial x]$ is a row vector whose ith component is $\partial V/\partial x_i$.

Lyapunov's stability theorem states that *the origin is stable if there is a continuously differentiable positive definite function $V(x)$ so that $\dot{V}(x)$ is negative semidefinite, and it is asymptotically stable if $\dot{V}(x)$ is negative definite*. A function $V(x)$ satisfying the conditions for stability is called a **Lyapunov function**. The surface $V(x) = c$, for some $c > 0$, is called a **Lyapunov surface** or a level surface.

When $\dot{V}(x)$ is only negative semidefinite, we may still conclude asymptotic stability of the origin if we can show that no solution can stay identically in the set $\{\dot{V}(x) = 0\}$, other than the zero solution $x(t) \equiv 0$. Under this condition, $V(x(t))$ must decrease toward 0, and consequently $x(t)$

converges to zero as t tends to infinity. This extension of the basic theorem is known as **the invariance principle**.

Lyapunov functions can be used to estimate the region of attraction of an asymptotically stable origin, that is, to find sets contained in the region of attraction. Let $V(x)$ be a Lyapunov function that satisfies the conditions of asymptotic stability over a domain D. For a positive constant c, let Ω_c be the component of $\{V(x) \leq c\}$ that contains the origin in its interior. The properties of V guarantee that, by choosing c small enough, Ω_c will be bounded and contained in D. Then, every trajectory starting in Ω_c remains in Ω_c and approaches the origin as $t \to \infty$. Thus, Ω_c is an estimate of the region of attraction. If $D = R^n$ and $V(x)$ is radially unbounded, that is, $\|x\| \to \infty$ implies that $V(x) \to \infty$, then any point $x \in R^n$ can be included in a bounded set Ω_c by choosing c large enough. Therefore, *the origin is globally asymptotically stable if there is a continuously differentiable, radially unbounded function $V(x)$ such that for all $x \in R^n$, $V(x)$ is positive definite and $\dot{V}(x)$ is either negative definite or negative semidefinite but no solution can stay identically in the set $\{\dot{V}(x) = 0\}$ other than the zero solution $x(t) \equiv 0$.*

Time-Varying Systems

Equation (1) is time-invariant because f does not depend on t. The more general time-varying system is represented by

$$\dot{x} = f(t, x) \tag{4}$$

In this case, we may allow the Lyapunov function candidate V to depend on t. Let $V(t, x)$ be a continuously differentiable function defined for all $t \geq 0$ and $x \in D$. The derivative of V along the trajectories of Eq. (4) is given by

$$\dot{V}(t, x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x)$$

If there are positive definite functions $W_1(x)$, $W_2(x)$, and $W_3(x)$ such that

$$W_1(x) \leq V(t, x) \leq W_2(x) \tag{5}$$

$$\dot{V}(t, x) \leq -W_3(x) \tag{6}$$

for all $t \geq 0$ and all $x \in D$, then the origin is uniformly asymptotically stable, where "uniformly" indicates that the ε–δ definition of stability and the convergence of $x(t)$ to zero are independent of the initial time t_0. Such uniformity annotation is not needed with time-invariant systems since the solution of a time-invariant state equation starting at time t_0 depends only on the difference $t - t_0$, which is not the case for time-varying systems. If the inequalities of Eqs. (5) and (6) hold globally and $W_1(x)$ is radially unbounded, then the origin is globally uniformly asymptotically stable. If $W_1(x) = k_1\|x\|^a$, $W_2(x) = k_2\|x\|^a$, and $W_3(x) = k_3\|x\|^a$ for some positive constants k_1, k_2, k_3, and a, then the origin is exponentially stable.

Perturbed Systems

Consider the system

$$\dot{x} = f(t, x) + g(t, x) \qquad (7)$$

where f and g are continuous in t and locally Lipschitz in x, for all $t \geq 0$ and $x \in D$, in which $D \subset R^n$ is a domain that contains the origin $x = 0$. Suppose $f(t, 0) = 0$ and $g(t, 0) = 0$ so that the origin is an equilibrium point of Eq. (7). We think of the system (7) as a perturbation of the nominal system (4). The perturbation term $g(t, x)$ could result from modeling errors, uncertainties, or disturbances. In a typical situation, we do not know $g(t, x)$, but we know some information about it, like knowing an upper bound on $\|g(t, x)\|$. Suppose the nominal system has an exponentially stable equilibrium point at the origin, what can we say about the stability of the origin as an equilibrium point of the perturbed system? A natural approach to address this question is to use a Lyapunov function for the nominal system as a Lyapunov function candidate for the perturbed system.

Let $V(t, x)$ be a Lyapunov function that satisfies

$$c_1\|x\|^2 \leq V(t, x) \leq c_2\|x\|^2 \qquad (8)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq -c_3\|x\|^2 \qquad (9)$$

$$\left\| \frac{\partial V}{\partial x} \right\| \leq c_4\|x\| \qquad (10)$$

for all $x \in D$ for some positive constants c_1, c_2, c_3, and c_4. Suppose the perturbation term $g(t, x)$ satisfies the linear growth bound

$$\|g(t, x)\| \leq \gamma\|x\|, \quad \forall\, t \geq 0, \ \forall\, x \in D \qquad (11)$$

where γ is a nonnegative constant. We use V as a Lyapunov function candidate to investigate the stability of the origin as an equilibrium point for the perturbed system. The derivative of V along the trajectories of Eq. (7) is given by

$$\dot{V}(t, x) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) + \frac{\partial V}{\partial x} g(t, x)$$

The first two terms on the right-hand side are the derivative of $V(t, x)$ along the trajectories of the nominal system, which is negative definite and satisfies the inequality of Eq. (9). The third term, $[\partial V/\partial x]g$, is the effect of the perturbation. Using Eqs. (9) through (11), we obtain

$$\dot{V}(t, x) \leq -c_3\|x\|^2 + \left\| \frac{\partial V}{\partial x} \right\| \|g(t, x)\|$$

$$\leq -c_3\|x\|^2 + c_4\gamma\|x\|^2$$

If $\gamma < c_3/c_4$, then

$$\leq -(c_3 - \gamma c_4)\|x\|^2, \ \ (c_3 - \gamma c_4) > 0$$

which shows that the origin is an exponentially stable equilibrium point of the perturbed system (7).

Summary

Lyapunov's method is a powerful tool for studying the stability of equilibrium points. However, there are two drawbacks of the method. First,

there is no systematic procedure for finding Lyapunov functions. Second, the conditions of the theory are only sufficient; they are not necessary. Failure of a Lyapunov function candidate to satisfy the conditions for stability or asymptotic stability does not mean that the origin is not stable or asymptotically stable. These drawbacks have been mitigated by a long history of using the method in the analysis and design of engineering systems, where various techniques for finding Lyapunov functions for specific systems have been determined.

Cross-References

▶ Feedback Stabilization of Nonlinear Systems
▶ Input-to-State Stability
▶ Regulation and Tracking of Nonlinear Systems

Recommended Reading

For an introduction to Lyapunov's stability theory at the level of first-year graduate students, the textbooks Khalil (2002), Sastry (1999), Slotine and Li (1991), and (Vidyasagar 2002) are recommended. The books by Bacciotti and Rosier (2005) and Haddad and Chellaboina (2008) cover a wider set of topics at the same introductory level. A deeper look into the theory is provided in the monographs Hahn (1967), Krasovskii (1963), Rouche et al.

(1977), Yoshizawa (1966), and (Zubov 1964). Lyapunov's theory for discrete-time systems is presented in Haddad and Chellaboina (2008) and Qu (1998). The monograph Michel and Wang (1995) presents Lyapunov's stability theory for general dynamical systems, including functional and partial differential equations.

Bibliography

Bacciotti A, Rosier L (2005) Liapunov functions and stability in control theory, 2nd edn. Springer, Berlin

Haddad WM, Chellaboina V (2008) Nonlinear dynamical systems and control. Princeton University Press, Princeton

Hahn W (1967) Stability of motion. Springer, New York

Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Princeton

Krasovskii NN (1963) Stability of motion. Stanford University Press, Stanford

Michel AN, Wang K (1995) Qualitative theory of dynamical systems. Marcel Dekker, New York

Qu Z (1998) Robust control of nonlinear uncertain systems. Wiley-Interscience, New York

Rouche N, Habets P, Laloy M (1977) Stability theory by Lyapunov's direct method. Springer, New York

Sastry S (1999) Nonlinear systems: analysis, stability, and control. Springer, New York

Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs

Vidyasagar M (2002) Nonlinear systems analysis, classic edn. SIAM, Philadelphia

Yoshizawa T (1966) Stability theory by Liapunov's second method. The Mathematical Society of Japan, Tokyo

Zubov VI (1964) Methods of A. M. Lyapunov and their applications. P. Noordhoff Ltd., Groningen

L

M

Markov Chains and Ranking Problems in Web Search

Hideaki Ishii[1] and Roberto Tempo[2]
[1]Tokyo Institute of Technology, Yokohama, Japan
[2]CNR-IEIIT, Politecnico di Torino, Torino, Italy

Abstract

Markov chains refer to stochastic processes whose states change according to transition probabilities determined only by the states of the previous time step. They have been crucial for modeling large-scale systems with random behavior in various fields such. as control, communications, biology, optimization, and economics. In this entry, we focus on their recent application to the area of search engines, namely, the PageRank algorithm employed at Google, which provides a measure of importance for each page in the web. We present several researches carried out with control theoretic tools such as aggregation, distributed randomized algorithms, and PageRank optimization. Due to the large size of the web, computational issues are the underlying motivation of these studies.

Keywords

Aggregation; Distributed randomized algorithms; Markov chains; Optimization; PageRank; Search engines; World wide web

Introduction

For various real-world large-scale dynamical systems, reasonable models describing highly complex behaviors can be expressed as stochastic systems, and one of the most well-studied classes of such systems is that of Markov chains. A characteristic feature of Markov chains is that their behavior does not carry any memory. That is, the current state of a chain is completely determined by the state of the previous time step and not at all on the states prior to that step (Kumar and Varaiya 1986; Norris 1997).

Recently, Markov chains have gained renewed interest due to the extremely successful applications in the area of web search. The search engine of Google has been employing an algorithm known as PageRank to assist the ranking of search results. This algorithm models the network of web pages as a Markov chain whose states represent the pages that web surfers with various interests visit in a random fashion. The objective is to find an order among the pages according to their

popularity and importance, and this is done by focusing on the structure of hyperlinks among pages.

In this entry, we first provide a brief overview on the basics of Markov chains and then introduce the problem of PageRank computation. We proceed to provide further discussions on control theoretic approaches dealing with PageRank problems. The topics covered include aggregated Markov chains, distributed randomized algorithms, and Markov decision problems for link optimization.

Markov Chains

In the simplest form, a Markov chain takes its states in a finite state space with transitions in the discrete-time domain. The transition from one state to another is characterized completely by the underlying probability distribution.

Let $\mathcal{X}$ be a finite set given by $\mathcal{X} := \{1, 2, \ldots, n\}$, which is called the state space. Consider a stochastic process $\{X_k\}_{k=0}^{\infty}$ taking values on this set $\mathcal{X}$. Such a process is called a Markov chain if it exhibits the following Markov property:

$$\text{Prob}\{X_{k+1} = j | X_k = i_k, \ X_{k-1} = i_{k-1}, \ldots, $$
$$X_0 = i_0\} = \text{Prob}\{X_{k+1} = j | X_k = i_k\},$$

where $\text{Prob}\{\cdot | \cdot\}$ denotes the conditional probability and $k \in \mathbb{Z}_+$. That is, the state at the next time step depends only on the current state and not those of previous times.

Here, we consider the homogeneous case where the transition probability is constant over time. Thus, we have for each pair $i, j \in \mathcal{X}$, the probability that the chain goes from state j to state i at time k expressed as

$$p_{ij} := \text{Prob}\{X_k = i | X_{k-1} = j\}, \ k \in \mathbb{Z}_+.$$

In the matrix form, $P := (p_{ij})$ is called the transition probability matrix of the chain. It is obvious that all entries of P are nonnegative, and for each j, the entries of the jth column of P

sum up to 1, i.e., $\sum_{i=1}^{n} p_{ij} = 1$ for $j \in \mathcal{X}$. In this respect, the matrix P is (column) stochastic (Horn and Johnson 1985).

In this entry, we assume that the Markov chain is ergodic, meaning that for any pair of states, the chain can make a transition from one to the other over time. In this case, the chain and the matrix P are called irreducible. This property is known to imply that P has a simple eigenvalue of 1. Thus, there exists a unique steady state probability distribution $\pi \in \mathbb{R}^n$ given by

$$\pi = P\pi, \ \ \mathbf{1}^T \pi = 1, \ \ \pi_i > 0, \ \forall i \in \mathcal{X},$$

where $\mathbf{1} \in \mathbb{R}^n$ denotes a vector with entries one. Note that in this distribution π, all entries are positive.

Ranking in Search Engines: PageRank Algorithm

At Google, PageRank is used to quantify the importance of each web page based on the hyperlink structure of the web (Brin and Page 1998; Langville and Meyer 2006). A page is considered important if (i) many pages have links pointing to the page, (ii) such pages having links are important ones, and (iii) the numbers of links that such pages have are limited. Intuitively, these requirements are reasonable. For a web page, its incoming links can be viewed as votes supporting the page, and moreover the quality of the votes count through their importance as well as the number of votes that they make. Even if a minor page (with low PageRank) has many outgoing links, its contribution to the linked pages will not be substantial.

An interesting way to explain the PageRank is through the *random surfer model*: The random surfer starts from a randomly chosen page. Each time visiting a page, he/she follows a hyperlink in that page chosen at random with uniform probability. Hence, if the current page i has n_i outgoing links, then one of them is picked with probability $1/n_i$. If it happens that the current page has no outgoing link (e.g., at PDF documents), the surfer will use the back button. This process

will be repeated. The PageRank value for a page represents the probability of the surfer visiting the page. It is thus higher for pages visited more often by the surfer.

It is now clear that PageRank is obtained by describing the random surfer model as a Markov chain and then finding its stationary distribution. First, we express the network of web pages as the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes corresponding to web page indices while $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges for links among pages. Node i is connected to node j by an edge, i.e., $(i, j) \in \mathcal{E}$, if page i has an outgoing link to page j.

Let $x_i(k)$ be the distribution of the random surfer visiting page i at time k, and let $x(k)$ be the vector containing all $x_i(k)$. Given the initial distribution $x(0)$, which is a probability vector, i.e., $\sum_{i=1}^{n} x_i(0) = 1$, the evolution of $x(k)$ can be expressed as

$$x(k + 1) = Ax(k). \tag{1}$$

The link matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is given by $a_{ij} = 1/n_j$ if $(j, i) \in \mathcal{E}$ and 0 otherwise, where n_j is the number of outgoing links of page j. Note that this matrix A is the transition probability matrix of the random surfer. Clearly, it is stochastic, and thus $x(k)$ remains a probability vector so that $\sum_{i=1}^{n} x_i(k) = 1$ for all k.

As mentioned above, PageRank is the stationary distribution of the process (1) under the assumption that the limit exists. Hence, the PageRank vector is given by $x^* := \lim_{k \to \infty} x(k)$. In other words, it is the solution of the linear equation

$$x^* = Ax^*, \quad x^* \in [0, 1]^n, \quad \mathbf{1}^T x^* = 1. \tag{2}$$

Notice that the PageRank vector x^* is a nonnegative unit eigenvector for the eigenvalue 1 of A. Such a vector exists since the matrix A is stochastic, but may not be unique; the reason is that A is a reducible matrix since in the web, not every pair of pages can be connected by simply following links. To resolve this issue, a slight modification is necessary in the random surfer model.

The idea of the *teleportation model* is that the random surfer, after a while, becomes bored and stops following the hyperlinks. At such an instant, the surfer "jumps" to another page not directly connected to the one currently visiting. This page can be in fact completely unrelated in the domains and/or the contents. All n pages in the web have the same probability $1/n$ to be reached by a jump.

The probability to make such a jump is denoted by $m \in (0, 1)$. The original transition probability matrix A is now replaced with the modified one $M \in \mathbb{R}^{n \times n}$ defined by

$$M := (1 - m)A + \frac{m}{n}\mathbf{1}\mathbf{1}^T. \tag{3}$$

For the value of m, we take $m = 0.15$ as reported in the original algorithm in Brin and Page (1998). Notice that M is a positive stochastic matrix. By Perron's theorem (Horn and Johnson 1985), the eigenvalue 1 is of multiplicity 1 and is the unique eigenvalue with the maximum modulus. Further, the corresponding eigenvector is positive. Hence, we redefine the vector x^* in (2) by using M instead of A as follows:

$$x^* = Mx^*, \quad x^* \in [0, 1]^n, \quad \mathbf{1}^T x_i^* = 1.$$

Due to the large dimension of the link matrix M, the computation of x^* is difficult. The solution employed in practice is based on the power method given by

$$x(k + 1) = Mx(k) = (1 - m)Ax(k) + \frac{m}{n}\mathbf{1}, \tag{4}$$

where the initial vector $x(0) \in \mathbb{R}^n$ is a probability vector. The second equality above follows from the fact $\mathbf{1}^T x(k) = 1$ for $k \in \mathbb{Z}_+$. For implementation, the form on the far right-hand side is important, using only the sparse matrix A and not the dense matrix M. This method asymptotically finds the value vector as $x(k) \to x^*, k \to \infty$.

Aggregation Methods for Large-Scale Markov Chains

In dealing with large-scale Markov chains, it is often desirable to predict their dynamic behaviors from reduced-order models that are more computationally tractable. This enables us, for example, to analyze the system performance at a macroscale with some approximation under different operating conditions. Aggregation refers to partitioning or grouping the states so that the states in each group can be treated as a whole. The technique of aggregation is especially effective for Markov chains possessing sparse structures with strong interactions among states in the same group and weak interactions among states in different groups. Such methods have been extensively studied, motivated by applications in queueing networks, power systems, etc. (Meyer 1989).

In the context of the PageRank problem, such sparse interconnection can be expressed in the link matrix A with a block-diagonal structure (after some coordinate change, if necessary). The entries of the matrix A are dense along its diagonal in blocks, and those outside the blocks take small values. More concretely, we write

$$A = I + B + \epsilon C, \qquad (5)$$

where B is a block-diagonal matrix given as $B = \mathrm{diag}(B_{11}, B_{22}, \ldots, B_{NN})$; B_{ii} is the $\tilde{n}_i \times \tilde{n}_i$ matrix corresponding to the ith group with $\tilde{n}_i$ member pages for $i = 1, 2, \ldots, N$; and ϵ is a small positive parameter. Here, the non-diagonal entries of B_{ii} are the same as those in the same diagonal block of A, but the diagonal entries are chosen such that $I + B_{ii}$ becomes stochastic and thus take nonpositive values. Thus, both B and C have column sums equal to zero. The small ϵ suggests us that states can be aggregated into N groups with strong interactions within the groups, but connections among different groups are weak. This class of Markov chains is known as nearly completely decomposable. In general, however, it is difficult to uniquely determine the form (5) for a given chain.

To exploit the sparse structure in the computation of stationary probability distributions, one approach is to carry out decomposition or aggregation of the chains. The basic approach here is (i) to compute the local stationary distributions for $I + B_{ii}$, (ii) to find the global stationary distribution for a chain representing the group interactions, and (iii) to finally use the obtained vectors to compute exact/approximate distribution for the entire chain; for details, see Meyer (1989). By interpreting such methods from the control theoretic viewpoints, in Phillips and Kokotovic (1981) and Aldhaheri and Khalil (1991), singular perturbation approaches have been developed. These methods lead us to the two-time scale decomposition of (controlled) Markov chain recursions.

In the case of PageRank computation, sparsity is a relevant property since it is well known that many links in the web are intra-host ones, connecting pages within the same domains or directories. However, in the real web, it is easy to find pages that have only a few outlinks, but some of them are external ones. Such pages will prevent the link matrix from having small ϵ when decomposed in the form (5). Hence, the general aggregation methods outlined above are not directly applicable.

An aggregation-based method suitable for PageRank computation is proposed in Ishii et al. (2012). There, the sparsity in the web is expressed by the limited number of external links pointing towards pages in other groups. For each page i, the node parameter $\delta_i \in [0, 1]$ is given by

$$\delta_i := \frac{\text{\# external outgoing links}}{\text{\# total outgoing links}}.$$

Note that smaller δ_i implies sparser networks. In this approach, for a given bound δ, the condition $\delta_i \leq \delta$ is imposed only in the case page i belongs to a group consisting of multiple members. Thus, a page forming a group by itself is not required to satisfy the condition. This means that we can regroup the pages by first identifying pages that violate this condition in the initial groups and then making them separately as *single* groups. By repeating these steps, it is always possible to

obtain groups for a given web. Once the grouping is settled, an aggregation-based algorithm can be applied, which computes an approximated PageRank vector. A characteristic feature is the tradeoff between the accuracy in PageRank computation and the node parameter δ. More accurate computation requires a larger number of groups and thus a smaller δ.

Distributed Randomized Computation

For large-scale computation, distributed algorithms can be effective by employing multiple processors to compute in parallel. There are several methods of constructing algorithms to find stationary distributions of large Markov chains. In this section, motivated by the current literature on multi-agent systems, sequential distributed randomized approaches of gossip type are described for the PageRank problem.

In *gossip-type* distributed algorithms, nodes make decisions and transmit information to their neighbors in a random fashion. That is, at any time instant, each node decides whether to communicate or not depending on a random variable. The random property is important to make the communication asynchronous so that simultaneous transmissions resulting in collisions can be avoided. Moreover, there is no need of any centralized decision maker or fixed order among pages.

More precisely, each page $i \in \mathcal{V}$ is equipped with a random process $\eta_i(k) \in \{0, 1\}$ for $k \in \mathbb{Z}_+$. If at time k, $\eta_i(k)$ is equal to 1, then page i broadcasts its information to its neighboring pages connected by outgoing links. All pages involved at this time renew their values based on the latest available data. Here, $\eta_i(k)$ is assumed to be an independent and identically distributed (i.i.d.) random process, and its probability distribution is given by Prob$\{\eta_i(k) = 1\} = \alpha$, $k \in \mathbb{Z}_+$. Hence, all pages are given the same probability α to initiate an update.

One of the proposed randomized approaches is based on the so-called asynchronous iteration algorithms for distributed computation of fixed points in the field of numerical analysis (Bertsekas and Tsitsiklis 1989). The distributed update recursion is given as

$$\check{x}(k+1) = \check{M}_{\eta_1(k),\ldots,\eta_n(k)}\check{x}(k), \qquad (6)$$

where the initial state $\check{x}(0)$ is a probability vector and the distributed link matrices $\check{M}_{p_1,\ldots,p_n}$ are given as follows: Its (i, j)th entry is equal to $(1-m)a_{ij} + m/n$ if $p_i = 1$; 1 if $p_i = 0$ and $i = j$; and 0 otherwise. Clearly, these matrices keep the rows of the original link matrix M in (3) for pages initiating updates. Other pages just keep their previous values. Thus, these matrices are not stochastic. From this update recursion, the PageRank x^* is probabilistically obtained (in the mean square sense and in probability one), where the convergence speed is exponential in time k. Note that in this scheme (6), due to the way the distributed link matrices are constructed, each page needs to know which pages have links pointing towards it. This implies that popular pages linked by a number of pages must have extra memory to keep the data of such links.

Another recently developed approach Ishii and Tempo (2010) and Zhao et al. (2013) has several notable differences from the asynchronous iteration approach above. First, the pages need to transmit their states only over their outgoing links; the information of such links are by default available locally, and thus, pages are not required to have the extra memory regarding incoming links. Second, it employs stochastic matrices in the update as in the centralized scheme; this aspect is utilized in the convergence analysis. As a consequence, it is established that the PageRank vector x^* is computed in a probabilistic sense through the time average of the states $x(0), \ldots, x(k)$ given by $y(k) := 1/(k+1) \sum_{\ell=0}^{k} x(\ell)$. The convergence speed in this case is of order $1/k$.

PageRank Optimization via Hyperlink Designs

For owners of websites, it is of particular interest to raise the PageRank values of their web pages. Especially in the area of e-business, this can

be critical for increasing the number of visitors to their sites. The values of PageRank can be affected by changing the structure of hyperlinks in the owned pages. Based on the random surfer model, intuitively, it makes sense to arrange the links so that surfers will stay within the domain of the owners as long as possible.

PageRank optimization problems have rigorously been considered in, for example, de Kerchove et al. (2008) and Fercoq et al. (2013). In general, these are combinatorial optimization problems since they deal with the issues on where to place hyperlinks, and thus the computation for solving them can be prohibitive especially when the web data is large. However, the work Fercoq et al. (2013) has shown that the problem can be solved in polynomial time. In what follows, we discuss a simplified discrete version of the problem setup of this work.

Consider a subset $\mathcal{V}_0 \subset \mathcal{V}$ of web pages over which a webmaster has control. The objective is to maximize the total PageRank of the pages in this set $\mathcal{V}_0$ by finding the outgoing links from these pages. Each page $i \in \mathcal{V}_0$ may have constraints such as links that must be placed within the page and those that cannot be allowed. All other links, i.e., those that one can decide to have or not, are the design parameters. Hence, the PageRank optimization problem can be stated as

$$\max\{U(x^*, M) : \ x^* = M x^*, \ x^* \in [0, 1]^n,$$
$$\mathbf{1}^T x^* = 1, \ M \in \mathcal{M}\},$$

where U is the utility function $U(x^*, M) := \sum_{i \in \mathcal{V}_0} x_i^*$ and $\mathcal{M}$ represents the set of admissible link matrices in accordance with the constraints introduced above.

In Fercoq et al. (2013), an extended continuous problem is also studied where the set $\mathcal{M}$ of link matrices is a polytope of stochastic matrices and a more general utility function is employed. The motivation for such a problem comes from having weighted links so that webmasters can determine which links should be placed in a more visible location inside their pages to increase clickings on those hyperlinks. Both discrete and continuous problems are shown to be solvable

in polynomial time by modeling them as constrained Markov decision processes with ergodic rewards (see, e.g., Puterman 1994).

Summary and Future Directions

Markov chains form one of the simplest classes of stochastic processes but have been found powerful in their capability to model large-scale complex systems. In this entry, we introduced them mainly from the viewpoint of PageRank algorithms in the area of search engines and with a particular emphasis on recent works carried out based on control theoretic tools. Computational issues will remain in this area as major challenges, and further studies will be needed. As we have observed in PageRank-related problems, it is important to pay careful attention to structures of the particular problems.

Cross-References

▶ Randomized Methods for Control of Uncertain Systems

Bibliography

Aldhaheri R, Khalil H (1991) Aggregation of the policy iteration method for nearly completely decomposable Markov chains. IEEE Trans Autom Control 36:178–187

Bertsekas D, Tsitsiklis J (1989) Parallel and distributed computation: numerical methods. Prentice-Hall, Englewood Cliffs

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30:107–117

de Kerchove C, Ninove L, Van Dooren P (2008) Influence of the outlinks of a page on its PageRank. Linear Algebra Appl 429:1254–1276

Fercoq O, Akian M, Bouhtou M, Gaubert S (2013) Ergodic control and polyhedral approaches to PageRank optimization. IEEE Trans Autom Control 58:134–148

Horn R, Johnson C (1985) Matrix analysis. Cambridge University Press, Cambridge

Ishii H, Tempo R (2010) Distributed randomized algorithms for the PageRank computation. IEEE Trans Autom Control 55:1987–2002

Ishii H, Tempo R, Bai EW (2012) A web aggregation approach for distributed randomized PageRank algorithms. IEEE Trans Autom Control 57:2703–2717

Kumar P, Varaiya P (1986) Stochastic systems: estimation, identification, and adaptive control. Prentice Hall, Englewood Cliffs

Langville A, Meyer C (2006) Google's PageRank and beyond: the science of search engine rankings. Princeton University Press, Princeton

Meyer C (1989) Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. SIAM Rev 31:240–272

Norris J (1997) Markov chains. Cambridge University Press, Cambridge

Phillips R, Kokotovic P (1981) A singular perturbation approach to modeling and control of Markov chains. IEEE Trans Autom Control 26:1087–1094

Puterman M (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, New York

Zhao W, Chen H, Fang H (2013) Convergence of distributed randomized PageRank algorithms. IEEE Trans Autom Control 58:3255–3259

Mathematical Models of Marine Vehicle-Manipulator Systems

Gianluca Antonelli
University of Cassino and Southern Lazio, Cassino, Italy

Abstract

Marine intervention requires the use of manipulators mounted on support vehicles. Such systems, defined as vehicle-manipulator systems, exhibit specific mathematical properties and require proper control design methodologies. This article briefly discusses the mathematical model within a control perspective as well as sensing and actuation peculiarities.

Keywords

Floating-base manipulators; Marine robotics; Underwater intervention; Underwater robotics

Introduction

In case of marine operations that require interaction with the environment, an underwater vehicle is usually equipped with one or more manipulators; such systems are defined underwater vehicle-manipulator systems (UVMSs). A UVMS holding six degree-of-freedom (DOF) manipulators is illustrated in Fig. 1. The vehicle carrying the manipulator may or may not be connected to the surface; in the first case we face a so-called remotely operated vehicle (ROV), while in the latter an autonomous underwater vehicle (AUV). ROVs, being physically linked, via the tether, to an operator that can be on a submarine or on a surface ship, receives power as well as control commands. AUVs, on the other hand, are supposed to be completely autonomous, thus relying to onboard power system and *intelligence*.

Remotely controlled UVMSs represent the state of the art in underwater manipulation, while autonomous or semiautonomous UVMSs still are in their embryonic stage. All over the world, few experimental setups have been developed within on-purpose projects; see, e.g., the European project Trident (2012).

Sensory System

Any manipulation task requires that some variables are measured; those may concern the internal state of the system such as the end effector as well the vehicle position and orientation or the velocities. Some others concern the surrounding environment as it is the case of vision systems or range measurements. Underwater sensing is characterized by poorer performance with respect to the ground corresponding variables due to the physical properties of the water as medium carrying the electromagnetic or acoustic signals.

One of the major challenges in underwater robotics is the localization due to the absence of a single, proprioceptive sensor that measures the vehicle position and the impossibility to use the Global Navigation Satellite System (GNSS) under the water. The use of redundant multisensor systems, thus, is common in order to perform sensor fusion and give fault detection and tolerance capabilities to the vehicle.
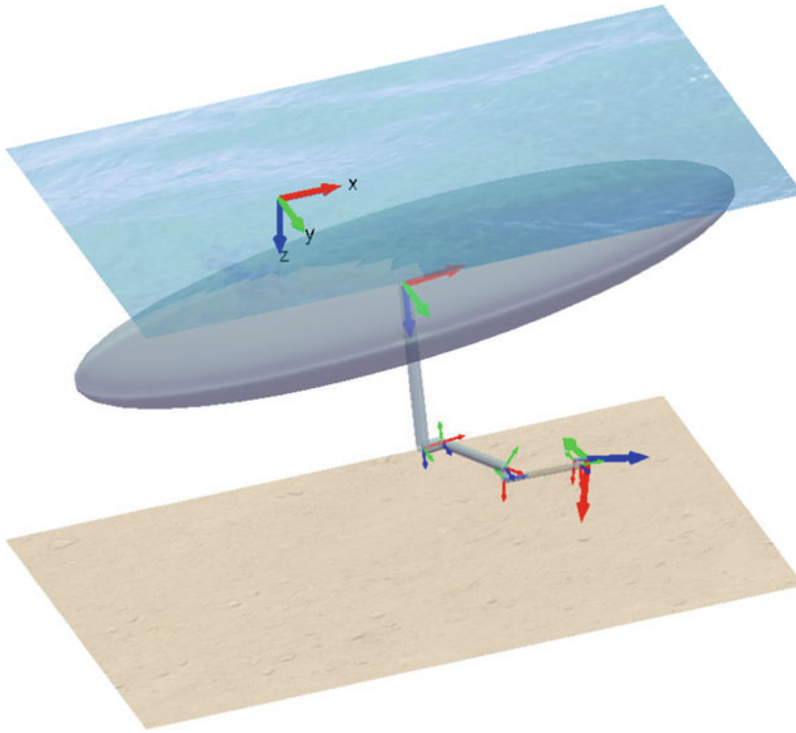
Mathematical Models of Marine Vehicle-Manipulator Systems, Fig. 1 Sketch of a UVMS, the inertial frame as well as the frames attached to all the rigid bodies are highlighted

Localization

A possible approach for AUV localization is to rely on inertial navigation systems (INSs); those are algorithms that implement dead reckoning techniques, i.e., the estimation of the position by properly merging and integrating measurements obtained with inertial and velocity sensors. Dead reckoning suffers from numerical drift due to the integration of sensor noise, as well as sensor bias and drift, and may be prone to the presence of external currents and model uncertainties. Since the variance of the estimation error grows with the distance traveled, this technique is only used for short dives.

Several algorithms are based on the concept of trilateration. The vehicle measures its distance with respect to known positions and properly uses this information by applying geometric-based formulas. Under the water, the technology for trilateration is not based on the electromagnetic field, due to the attenuation of its radiations, but on acoustics.

Among the commercially available solutions, long, short, and ultrashort baseline systems have found widespread use. The differences are in the baseline wavelength, the required distance among the transponders, the accuracy, and the installation cost. Acoustic underwater positioning is commercially mature, and several companies offer a variety of products.

In case of intervention, when the UVMS is close to the target, rather than the absolute position with respect to an inertial frame, it is crucial to estimate the relative position with respect to the target itself. In such a case, vision-based systems may be considered.

Actuation

Underwater vehicles are usually controlled by thrusters and/or control surfaces. Control surfaces, such as rudders and sterns, are typically used in vehicles working at cruise speed, i.e., torpedo-shaped vehicles usually used in

monitoring or cable/pipeline inspection. In such vehicles a main thruster is used together with at least one rudder and one stern.

This configuration is unsuitable for UVMSs since the force/moment provided by the control surfaces is the function of the velocity and it is null in hovering, when typically manipulation is performed.

The relationship between the force/moment acting on the vehicle and the control input of the thrusters is highly nonlinear. It is the function of structural variables such as the density of the water, the tunnel cross-sectional area, the tunnel length, the volumetric flow rate between input and output of the thrusters, and the propeller diameter. The state of the dynamic system describing the thrusters is constituted by the propeller revolution, the speed of the fluid going into the propeller, and the input torque.

Modeling

UVMSs can be modeled as rigid bodies connected to form a serial chain; the vehicle is the floating base, while each link of the manipulator represents an additional rigid body with one DOF, typically the rotation around the corresponding joint's axis. Roughly speaking, modeling of a UVMS is the effort to represent the physical relationships of those bodies in order to measure and control its end effector, typically involved in a manipulation task.

The first step of modeling is the so-called direct kinematics, consisting in computing the position/orientation of the end effector with respect to an inertial, i.e., world fixed, frame. This is done via geometric relationship function of the system kinematic parameters, typically the lengths of the links, and the current system configuration, i.e., the vehicle position/orientation and the joint positions.

Velocities of each rigid body affect the following rigid bodies and thus the end effector. For example, a vehicle roll movement or the joint velocity is projected into a linear and angular end-effector velocity. This velocity transformation is studied by the differential kinematics.

Analytic and/or geometric approaches may be used to retrieve those relationships. The study of the velocity-related equations is fundamental to understand how to balance the movement between vehicle and manipulator and, within the manipulator, how to distribute it among the joints. This topic is strictly related to differential, and inverse, kinematics for industrial robots.

The extension of Newton's second law to UVMSs leads to a number of nonlinear differential equations that link together the systems generalized forces and accelerations. With the word generalized forces, it is here intended as the forces and moments acting on the vehicles and the joint torques. Correspondingly, one is interested in the vehicle linear and angular accelerations and joint accelerations. Those equations couple together all the DOFs of the structure, e.g., a force applied to the vehicle causes acceleration also on the joints. Study of the dynamics is crucial to design the controller.

It is not possible to neglect that the bodies are moving in the water, the theory of fluidodynamics is rather complex, and it is difficult to develop a *simple* model for most of the hydrodynamic effects. A rigorous analysis for incompressible fluids would need to resort to the Navier-Stokes equations (distributed fluid flow). However, most of the hydrodynamic effects have no significant influence in the range of the operative velocities for UVMS intervention tasks. In particular, it is necessary to model added masses, linear and quadratic damping terms, and the buoyancy.

Control

Not surprisingly, the mathematical model of UVMS shares most of the characteristics of industrial robots as well as space robots modeling. Having taken into account to the physical differences, the control problems are also similar:

- Kinematic control. The control problem is given in terms of motion of the end effector and needs to be transformed into the motion of the vehicle and the manipulator. This is often approached by resorting to the inverse

differential kinematic algorithms. In particular, algorithms for redundant systems need to be considered since a UVMS always possess at least six DOFs. Moving a UVMS requires to handle additional variables with respect to the end effector such as the vehicle roll and pitch to preserve energy, the robot manipulability, the mechanical joint limits, or eventual directional sensing.

- Motion control. Low-level control algorithms are designed to allow the system tracking the desired trajectory. UVMSs are characterized by different dynamics between vehicle and manipulator, uncertainty in the model parameter knowledge, poor sensing performance, and limit cycle in the thruster model. On the other hand, the limited bandwidth of the closed-loop system allows the use of simple control approaches.

- Interaction control. Several applications require exchange of forces with the environment. A pure motion control algorithm is not devised for such operation and specific force control algorithms, both direct and indirect, may be necessary. Master/slave systems or haptic devices may be used on the purpose, while autonomous interaction control still is in the research phase for the marine environment.

Summary and Future Directions

This article is aimed at giving a short overview of the main mathematical and technological challenges arising with UVMSs. All the components of an underwater mission, perception, actuation, and communication with the surface, are characterized by poorer performances with respect to the current industrial or advanced robotics applications.

The underwater environment is hostile; as an example the marine current provides disturbances to be counteracted by the dynamic controller, or the sand's whirlwinds obfuscate the vision-based operations close to the sea bottom. Both tele-operated and autonomous underwater missions require a significant human effort in planning,

testing, and monitoring all the operations. Fault detection and recovery policies are necessary in each step to avoid loss of expensive hardware.

Future generation of UVMSs needs to be autonomous, to percept and contextualize the environment, to react with respect to unplanned situations, and to safely reschedule the tasks of complex missions. Those characteristics are being shared by all the branches of the service robotics.

Cross-References

▸ Advanced Manipulation for Underwater Sampling
▸ Mathematical Models of Ships and Underwater Vehicles
▸ Redundant Robots

Recommended Reading

The book of Fossen (1994) is one of the first books dedicated to control problems of marine systems, both underwater and surface. The same author presents, in Fossen (2002), an updated and extended version of the topics developed in the first book and in Fossen (2011), a handbook on marine craft hydrodynamics and control. A short introductory chapter to marine robotics may be found in Antonelli et al. (2008). Robotics fundamentals are also useful and can be found in Siciliano et al. (2009). To the best of our knowledge, Antonelli (2014) is the only monograph devoted to addressing the specific problems of underwater vehicle-manipulator systems.

Bibliography

Antonelli, G (2014) Underwater robots: motion and force control of vehicle-manipulator systems. Springer tracts in advanced robotics, Springer-Verlag, 3rd edn. Springer, Heidelberg

Antonelli G, Fossen T, Yoerger D (2008) Underwater robotics. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Heidelberg, pp 987–1008

Fossen T (1994) Guidance and control of ocean vehicles. Chichester, New York

Fossen T (2002) Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles. Marine Cybernetics, Trondheim

Fossen T (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics: modelling, planning and control. Springer, London

TRIDENT (2012) Marine robots and dexterous manipulation for enabling autonomous underwater multipurpose intervention missions. http://www.irs.uji.es/trident

Mathematical Models of Ships and Underwater Vehicles

Thor I. Fossen
Department of Engineering Cyberentics, Centre for Autonomous Marine Operations and Systems, Norwegian University of Science and Technology, Trondheim, Norway

Abstract

This entry describes the equations of motion of ships and underwater vehicles. Standard hydrodynamic models in the literature are reviewed and presented using the nonlinear robot-like vectorial notation of Fossen (1991, 1994, 2011). The matrix-vector notation is highly advantageous when designing control systems since well-known system properties such as symmetry, skew-symmetry, and positiveness can be exploited in the design.

Keywords

Autonomous underwater vehicle (AUV); Degrees of freedom; Euler angles; Hydrodynamics; Kinematics; Kinetics; Maneuvering; Remotely operated vehicle (ROV); Seakeeping; Ship; Underwater vehicles

Introduction

The subject of this entry is mathematical modeling of ships and underwater vehicles. With *ship* we mean "any large floating vessel capable of crossing open waters," as opposed to a boat, which is generally a smaller craft. An *underwater vehicle* is a "small vehicle that is capable of propelling itself beneath the water surface as well as on the water's surface." This includes unmanned underwater vehicles (UUVs), remotely operated vehicles (ROVs), autonomous underwater vehicles (AUVs) and underwater robotic vehicles (URVs).

This entry is based on Fossen (1991, 2011), which contains a large number of standard models for ships, rigs, and underwater vehicles. There exist a large number of textbooks on mathematical modeling of ships; see Rawson and Tupper (1994), Lewanddowski (2004), and Perez (2005). For underwater vehicles, see Allmendinger (1990), Sutton and Roberts (2006), Inzartsev (2009), Antonelli (2010), and Wadoo and Kachroo (2010). Some useful references on ship hydrodynamics are Newman (1977), Faltinsen (1990), and Bertram (2012).

Degrees of Freedom

A mathematical model of marine craft is usually represented by a set of ordinary differential equations (ODEs) describing the motions in six degrees of freedom (DOF): *surge, sway, heave, roll, pitch,* and *yaw*.

Hydrodynamics

In hydrodynamics it is common to distinguish between two theories:

- **Seakeeping theory:** The motions of ships at zero or constant speed in waves are analyzed using hydrodynamic coefficients and wave forces, which depends of the wave excitation frequency and thus the hull geometry and mass distribution. For underwater vehicles operating below the wave-affected zone, the wave excitation frequency will not influence the hydrodynamic coefficients.

- **Maneuvering theory:** The ship is moving in restricted calm water – that is, in sheltered waters or in a harbor. Hence, the maneuvering model is derived for a ship moving at positive speed under a zero-frequency wave excitation

assumption such that added mass and damping can be represented by constant parameters.

Seakeeping models are typically used for ocean structures and dynamically positioned vessels. Several hundred ODEs are needed to effectively represent a seakeeping model; see Fossen (2011), and Perez and Fossen (2011a,b).

The remainder of this entry assumes *maneuvering theory*, since this gives lower-order models typically suited for controller-observer design. Six ODEs are needed to describe the *kinematics*, that is, the geometrical aspects of motion while Newton-Euler's equations represent additional six ODEs describing the forces and moments causing the motion (*kinetics*).

Notation

The equations of motion are usually represented using generalized position, velocity and forces (Fossen 1991, 1994, 2011) defined by the state vectors:

$$\boldsymbol{\eta} := [x, y, z, \phi, \theta, \psi]^\top \tag{1}$$

$$\boldsymbol{v} := [u, v, w, p, q, r]^\top \tag{2}$$

$$\boldsymbol{\tau} := [X, Y, Z, K, M, N]^\top \tag{3}$$

where $\boldsymbol{\eta}$ is the generalized position expressed in the north-east-down (NED) reference frame $\{n\}$. A body-fixed reference frame $\{b\}$ with axes:

x_b – longitudinal axis (from aft to fore)
y_b – transversal axis (to starboard)
z_b – normal axis (directed downward)
is rotating about the NED reference frame $\{n\}$ with angular velocity $\boldsymbol{\omega} = [p, q, r]^\top$. The generalized velocity vector $\boldsymbol{v}$ and forces $\boldsymbol{\tau}$ are both expressed in $\{b\}$, and the 6-DOF states are defined according to SNAME (1950):

- **Surge** position x, linear velocity u, force X
- **Sway** position y, linear velocity v, force Y
- **Heave** position z, linear velocity w, force Z
- **Roll** angle ϕ, angular velocity p, moment K
- **Pitch** angle θ, angular velocity q, moment M
- **Yaw** angle ψ, angular velocity r, moment N

Kinematics

The generalized velocities $\dot{\boldsymbol{\eta}}$ and $\boldsymbol{v}$ in $\{b\}$ and $\{n\}$, respectively satisfy the following kinematic transformation (Fossen 1994, 2011):

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{v} \tag{4}$$

$$\mathbf{J}(\boldsymbol{\eta}) := \begin{bmatrix} \mathbf{R}(\boldsymbol{\Theta}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{T}(\boldsymbol{\Theta}) \end{bmatrix} \tag{5}$$

where $\boldsymbol{\Theta} = [\phi, \theta, \psi]^\top$ is the *Euler angles* and

$$\mathbf{R}(\boldsymbol{\Theta}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi \\ -s\theta & c\theta s\phi \end{bmatrix}$$
$$\begin{bmatrix} s\psi s\phi + c\psi c\phi s\theta \\ -c\psi s\phi + s\theta s\psi c\phi \\ c\theta c\phi \end{bmatrix} \tag{6}$$

with $s\cdot := \sin(\cdot)$, $c\cdot := \cos(\cdot)$ and $t\cdot := \tan(\cdot)$.

The matrix $\mathbf{R}$ is recognized as the Euler angle rotation matrix $\mathbf{R} \in SO(3)$ satisfying $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$, and $\det(\mathbf{R}) = 1$, which implies that $\mathbf{R}$ is orthogonal. Consequently, the inverse rotation matrix is given by: $\mathbf{R}^{-1} = \mathbf{R}^\top$. The Euler rates $\dot{\boldsymbol{\Theta}} = T(\boldsymbol{\Theta})\boldsymbol{\omega}$ are singular for $\theta \neq \pm\pi/2$ since:

$$\mathbf{T}(\boldsymbol{\Theta}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \quad \theta \neq \pm\frac{\pi}{2} \tag{7}$$

Singularities can be avoided by using *unit quaternions* instead (Fossen 1994, 2011).

Kinetics

The rigid-body kinetics can be derived using the *Newton-Euler formulation*, which is based on *Newton's second law.* Following Fossen (1994, 2011) this gives:

$$\mathbf{M}_{RB}\dot{\boldsymbol{v}} + \mathbf{C}_{RB}(\boldsymbol{v})\boldsymbol{v} = \boldsymbol{\tau}_{RB} \tag{8}$$

where $\mathbf{M}_{RB}$ is the rigid-body mass matrix, $\mathbf{C}_{RB}$ is the rigid-body Coriolis and centripetal matrix due to the rotation of $\{b\}$ about the geographical frame $\{n\}$. The generalized force vector $\boldsymbol{\tau}_{RB}$ represents external forces and moments expressed in $\{b\}$. In the nonlinear case:

$$\boldsymbol{\tau}_{RB} = -\mathbf{M}_A\dot{\boldsymbol{v}} - \mathbf{C}_A(\boldsymbol{v})\boldsymbol{v} - \mathbf{D}(\boldsymbol{v})\boldsymbol{v} - \mathbf{g}(\boldsymbol{\eta}) + \boldsymbol{\tau} \tag{9}$$

where the matrices $\mathbf{M}_A$ and $\mathbf{C}_A(\mathbf{v})$ represent hydrodynamic added mass due to acceleration $\dot{\mathbf{v}}$ and Coriolis acceleration due to the rotation of $\{b\}$ about the geographical frame $\{n\}$. The potential and viscous damping terms are lumped together into a nonlinear matrix $\mathbf{D}(\mathbf{v})$ while $\mathbf{g}(\boldsymbol{\eta})$ is a vector of generalized restoring forces. The control inputs are generalized forces given by $\boldsymbol{\tau}$.

Formulae (8) and (9) together with (4) are the fundamental equations when deriving the ship and underwater vehicle models. This is the topic for the next sections.

Ship Model

The ship equations of motion are usually represented in three DOFs by neglecting *heave, roll* and *pitch*. Combining (4), (8), and (9) we get:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \qquad (10)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \qquad (11)$$

where $\boldsymbol{\eta} := [x, y, \psi]^\top$, $\mathbf{v} := [u, v, r]^\top$ and

$$\mathbf{R}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (12)$$

is the rotation matrix in yaw. It is assumed that wind and wave-induced forces $\boldsymbol{\tau}_{\text{wind}}$ and $\boldsymbol{\tau}_{\text{wave}}$ can be linearly superpositioned. The system matrices $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ and $\mathbf{C}(\mathbf{v}) = \mathbf{C}_{RB}(\mathbf{v}) + \mathbf{C}_A(\mathbf{v})$ are usually derived under the assumption of port-starboard symmetry and that surge can be decoupled from sway and yaw (Fossen 2011). Moreover,

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \qquad (13)$$

$$\mathbf{C}_{RB}(\mathbf{v}) = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix} \qquad (14)$$

$$\mathbf{C}_A(\mathbf{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \qquad (15)$$

Hydrodynamic damping will, in its simplest form, be linear:

$$\mathbf{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_r & -N_r \end{bmatrix} \qquad (16)$$

while a nonlinear expression based on second-order modulus functions describing quadratic drag and cross-flow drag is:

$$\mathbf{D}(\mathbf{v}) = \begin{bmatrix} -X_{|u|u}|u| & 0 \\ 0 & -Y_{|v|v}|v| - Y_{|r|v}|r| \\ 0 & -N_{|v|v}|v| - N_{|r|v}|r| \end{bmatrix}$$
$$\begin{matrix} 0 \\ -Y_{|v|r}|v| - Y_{|r|r}|r| \\ -N_{|v|r}|v| - N_{|r|r}|r| \end{matrix} \qquad (17)$$

Other nonlinear representations are found in Fossen (1994, 2011).

In the case of *irrotational ocean currents*, we introduce the relative velocity vector:

$$\mathbf{v}_r = \mathbf{v} - \mathbf{v}_c$$

where $\mathbf{v}_c = [u_c^b, v_c^b, 0]^\top$ is a vector of current velocities in $\{b\}$. Hence, the kinetic model takes the form:

$$\underbrace{\mathbf{M}_{RB}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v}}_{\text{rigid-body forces}}$$
$$+ \underbrace{\mathbf{M}_A\dot{\mathbf{v}}_r + \mathbf{C}_A(\mathbf{v}_r)\mathbf{v}_r + \mathbf{D}(\mathbf{v}_r)\mathbf{v}_r}_{\text{hydrodynamic forces}}$$
$$= \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}}$$

M

This model can be simplified if the *ocean currents* are assumed to be *constant* and *irrotational* in $\{n\}$. According to Fossen (2011, Property 8.1), $\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} \equiv \mathbf{M}_{RB}\dot{\boldsymbol{\nu}}_r + \mathbf{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ if the rigid-body Coriolis and centripetal matrix satisfies $\mathbf{C}_{RB}(\boldsymbol{\nu}_r) = \mathbf{C}_{RB}(\boldsymbol{\nu})$. One parametrization satisfying this is (14). Hence, the Coriolis and centripetal matrix satisfies $\mathbf{C}(\boldsymbol{\nu}_r) = \mathbf{C}_{RB}(\boldsymbol{\nu}_r) + \mathbf{C}_A(\boldsymbol{\nu}_r)$ and it follows that:

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \tag{18}$$

The kinematic equation (10) can be modified to include the relative velocity $\boldsymbol{\nu}_r$ according to:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu}_r + \left[u_c^n, v_c^n, 0\right]^\top \tag{19}$$

where the ocean current velocities $u_c^n = constant$ and $v_c^n = constant$ in $\{n\}$. Notice that the body-fixed velocities $\boldsymbol{\nu}_c = \mathbf{R}(\psi)^\top [u_c^n, v_c^n, 0]^\top$ will vary with the heading angle ψ.

The maneuvering model presented in this entry is intended for controller-observer design, prediction, and computer simulations, as well as system identification and parameter estimation. A large number of application-specific models for marine craft are found in Fossen (2011, Chap. 7).

Hydrodynamic programs compute mass, inertia, potential damping and restoringforces while

a more detailed treatment of viscous dissipative forces (damping) and sealoads are found in the extensive literature on hydrodynamics – see Faltinsen (1990) and Newman (1977).

Underwater Vehicle Model

The 6-DOF underwater vehicle equations of motion follow from (4), (8), and (9) under the assumption that wave-induced motions can be neglected:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{20}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{21}$$

with generalized position $\boldsymbol{\eta} := [x, y, z, \phi, \theta, \psi]^\top$ and velocity $\boldsymbol{\nu} := [u, v, w, p, q, r]^\top$. Assume that the gravitational force acts through the center of gravity (CG) defined by the vector $\mathbf{r}_g := [x_g, y_g, z_g]^\top$ with respect to the coordinate origin $\{b\}$. Similarly, the buoyancy force acts through the center of buoyancy (CB) defined by the vector $\mathbf{r}_b := [x_b, y_b, z_b]^\top$. For most vehicles $y_g = y_b = 0$.

For a port-starboard symmetrical vehicle with homogenous mass distribution, CG satisfying $y_g = 0$ and products of inertia $I_{xy} = I_{yz} = 0$, the system inertia matrix becomes:

$$\mathbf{M} := \begin{bmatrix} m - X_{\dot{u}} & 0 & -X_{\dot{w}} & 0 & mz_g - X_{\dot{q}} & 0 \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g - Y_{\dot{p}} & 0 & mx_g - Y_{\dot{r}} \\ -X_{\dot{w}} & 0 & m - Z_{\dot{w}} & 0 & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g - Y_{\dot{p}} & 0 & I_x - K_{\dot{p}} & 0 & -I_{zx} - K_{\dot{r}} \\ mz_g - X_{\dot{q}} & 0 & -mx_g - Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & mx_g - Y_{\dot{r}} & 0 & -I_{zx} - K_{\dot{r}} & 0 & I_z - N_{\dot{r}} \end{bmatrix} \tag{22}$$

where the hydrodynamic derivatives are defined according to SNAME (1950). The Coriolis and centripetal matrices are:

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix}$$

(23)

where

$$\begin{aligned} a_1 &= X_{\dot{u}}u + X_{\dot{w}}w + X_{\dot{q}}q \\ a_2 &= Y_{\dot{v}}v + Y_{\dot{p}}p + Y_{\dot{r}}r \\ a_3 &= Z_{\dot{u}}u + Z_{\dot{w}}w + Z_{\dot{q}}q \\ b_1 &= K_{\dot{v}}v + K_{\dot{p}}p + K_{\dot{r}}r \\ b_2 &= M_{\dot{u}}u + M_{\dot{w}}w + M_{\dot{q}}q \\ b_3 &= N_{\dot{v}}v + N_{\dot{p}}p + N_{\dot{r}}r \end{aligned}$$

(24)

and

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & -mr & mq & mz_g r & -mx_g q & -mx_g r \\ mr & 0 & -mp & 0 & m(z_g r + x_g p) & 0 \\ -mq & mp & 0 & -mz_g p & -mz_g q & mx_g p \\ -mz_g r & 0 & mz_g p & 0 & -I_{xz}p + I_z r & -I_y q \\ mx_g q & -m(z_g r + x_g p) & mz_g q & I_{xz}p - I_z r & 0 & -I_{xz}r + I_x p \\ mx_g r & 0 & -mx_g p & I_y q & I_{xz}r - I_x p & 0 \end{bmatrix}$$

(25)

Notice that this representation of $\mathbf{C}_{RB}(\boldsymbol{\nu})$ only depends on the angular velocities p, q, and r, and not the linear velocities $u, v,$ and r. This property will be exploited when including drift due to ocean currents.

Linear damping for a port-starboard symmetrical vehicle takes the following form:

$$\mathbf{D} = - \begin{bmatrix} X_u & 0 & X_w & 0 & X_q & 0 \\ 0 & Y_v & 0 & Y_p & 0 & Y_r \\ Z_u & 0 & Z_w & 0 & Z_q & 0 \\ 0 & K_v & 0 & K_p & 0 & K_r \\ M_u & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & N_p & 0 & N_r \end{bmatrix}$$

(26)

Let $W = mg$ and $B = \rho g \nabla$ denote the weight and buoyance where m is the mass of the vehicle including water in free floating space, ∇ the volume of fluid displaced by the vehicle, g the acceleration of gravity (positive downward), and ρ the water density. Hence, the generalized restoring forces for a vehicle satisfying $y_g = y_b = 0$ becomes (Fossen 1994, 2011):

$$\mathbf{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W - B)s\theta \\ -(W - B)c\theta s\phi \\ -(W - B)c\theta c\phi \\ (z_g W - z_b B)c\theta s\phi \\ (z_g W - z_b B)s\theta + (x_g W - x_b B)c\theta c\phi \\ -(x_g W - x_b B)c\theta s\phi \end{bmatrix}$$

(27)

The expression for $\mathbf{D}$ can be extended to include nonlinear damping terms if necessary. Quadratic damping is important at higher speeds since the Coriolis and centripetal terms $\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu}$ can destabilize the system if only linear damping is used.

In the presence of irrotational ocean currents, we can rewrite (20) and (21) in terms of relative velocity $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ according to:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu}_r + \left[u_c^n, v_c^n, w_c^n, 0, 0, 0\right]^\top \quad (28)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (29)$$

where it is assumed that $\mathbf{C}_{RB}(\boldsymbol{\nu}_r) = \mathbf{C}_{RB}(\boldsymbol{\nu})$, which clearly is satisfied for (25). In addition, it is assumed that u_c^n, v_c^n, and w_c^n are constant. For more details see Fossen (2011).

Programs and Data

The Marine Systems Simulator (MSS) is a Matlab/Simulink library and simulator for marine craft (http://www.marinecontrol.org). It includes models for ships, underwater vehicles, and floating structures.

Summary and Future Directions

This entry has presented standard models for simulation of ships and underwater vehicles. It is recommended to consult Fossen (1994, 2011) for a more detailed description of marine craft hydrodynamics.

Cross-References

▶ Control of Networks of Underwater Vehicles
▶ Control of Ship Roll Motion
▶ Dynamic Positioning Control Systems for Ships and Underwater Vehicles
▶ Underactuated Marine Control Systems

Bibliography

Allmendinger E (ed) (1990) Submersible vehicle systems design. SNAME, Jersey City
Anotonelli G (2010) Underwater robots: motion and force control of vehicle-manipulator systems. Springer, Berlin/New York
Bertram V (2012) Practical ship hydrodynamics. Elsevier, Amsterdam/London
Faltinsen O (1990) Sea loads on ships and offshore structures. Cambridge
Fossen TI (1991) Nonlinear modelling and control of underwater vehicles. PhD thesis, Department of Engineering Cybernetic, Norwegian University of Science and Technology
Fossen TI (1994) Guidance and control of ocean vehicles. Wiley, Chichester/New York
Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester/Hoboken
Inzartsev AV (2009) Intelligent underwater vehicles. I-Tech Education and Publishing. Open access: http://www.intechweb.org/
Lewanddowski EM (2004) The dynamics of marine craft. World Scientific, Singapore
MSS (2010) Marine systems simulator. Open access: http://www.marinecontrol.org/
Newman JN (1977) Marine hydrodynamics. MIT, Cambridge
Perez T (2005) Ship motion control. Springer, Berlin/London
Perez T, Fossen TI (2011a) Practical aspects of frequency-domain identification of dynamic models of marine structures from hydrodynamic data. Ocean Eng 38:426–435
Perez T, Fossen TI (2011b) Motion control of marine craft, Ch. 33. In: Levine WS (ed) The control systems handbook: control system advanced methods, 2nd edn. CRC, Hoboken
Rawson KJ, Tupper EC (1994) Basic ship theory. Longman Group, Harlow/New York
SNAME (1950) Nomenclature for treating the motion of a submerged body through a fluid. SNAME, Technical and Research Bulletin no 1–5, pp 1–15
Sutton R, Roberts G (2006) Advances in unmanned marine vehicles. IEE control series. The Institution of Engineering and Technology, London
Wadoo S, Kachroo P (2010) Autonomous underwater vehicles: modeling, control design and simulation. Taylor and Francis, London

Mean Field Games

Peter E. Caines
McGill University, Montreal, QC, Canada

Abstract

Mean Field Game (MFG) theory studies the existence of Nash equilibria, together with the individual strategies which generate them, in games involving a large number of agents modeled by controlled stochastic dynamical systems. This is achieved by exploiting the relationship between the finite and corresponding infinite limit population problems. The solution of the infinite population problem is given by the fundamental MFG Hamilton-Jacobi-Bellman (HJB) and Fokker-Planck-Kolmogorov (FPK) equations which are linked by the state distribution of a generic agent, otherwise known as the system's mean field.

Keywords

Fokker-Planck-Kolmogorov (FPK) equation; Hamilton-Jacobi-Bellman (HJB) equation; Nash equilibrium; Stochastic dynamical systems

Introduction

Large-population, dynamical, multi-agent, competitive, and cooperative phenomena occur in a wide range of designed and natural settings such as communication, environmental, epidemiological, transportation, and energy systems, and they underlie much economic and financial behavior. Analysis of such systems is intractable using the finite population game theoretic methods which have been developed for multi-agent control systems (see, e.g., Basar and Ho 1974; Basar and Olsder 1999; Ho 1980; and Bensoussan and Frehse 1984). The continuum population game theoretic models of economics (Aumann and Shapley 1974; Neyman 2002) are static, as, in general, are the large-population models employed in network games (Altman et al. 2002) and transportation analysis (Correa and Stier-Moses 2010; Haurie and Marcotte 1985; Wardrop 1952). However, dynamical (or sequential) stochastic games were analyzed in the continuum limit in the work of Jovanovic and Rosenthal (1988) and Bergin and Bernhardt (1992), where the fundamental mean field equations appear in the form of a discrete time dynamic programming equation and an evolution equation for the population state distribution.

The mean field equations for dynamical games with large but finite populations of asymptotically negligible agents originated in the work of Huang et al. (2003, 2006, 2007) (where the framework was called the Nash Certainty Equivalence Principle) and independently in that of Lasry and Lions (2006a,b, 2007), where the now standard terminology of Mean Field Games (MFGs) was introduced. Independent of both of these, the closely related notion of Oblivious

Equilibria for large-population dynamic games was introduced by Weintraub et al. (2005) in the framework of Markov Decision Processes (MDPs).

One of the main results of MFG theory is that in large-population stochastic dynamic games individual feedback strategies exist for which any given agent will be in a Nash equilibrium with respect to the pre-computable behavior of the mass of the other agents; this holds exactly in the asymptotic limit of an infinite population and with increasing accuracy for a finite population of agents using the infinite population feedback laws as the finite population size tends to infinity, a situation which is termed an ε-Nash equilibrium. This behavior is described by the solution to the infinite population MFG equations which are fundamental to the theory; they consist of (i) a parameterized family of HJB equations (in the nonuniform parameterized agent case) and (ii) a corresponding family of McKean-Vlasov (MV) FPK PDEs, where (i) and (ii) are linked by the probability distribution of the state of a generic agent, that is to say, the mean field. For each agent, these yield (i) a Nash value of the game, (ii) the best response strategy for the agent, (iii) the agent's stochastic differential equation (SDE) (i.e., the MV-SDE pathwise description), and (iv) the state distribution of such an agent (via the MV FPK for the parameterized individual).

Dynamical Agents

In the diffusion-based models of large-population games, the state evolution of a collection of N agents $A_i, 1 \leq i \leq N < \infty$, is specified by a set N of controlled stochastic differential equations (SDEs) which in the important linear case take the form

$$dx_i(t) = [F_i x_i(t) + G_i u_i(t)]dt + D_i dw_i(t),$$
$$1 \leq i \leq N,$$

where $x_i \in \mathbb{R}^n$ is the state, $u_i \in \mathbb{R}^m$ the control input, and w_i the state Wiener process of the ith

agent A_i, where $\{w_i, 1 \leq i \leq N\}$ is a collection of N independent standard Wiener processes in $\mathbb{R}^r$ independent of all mutually independent initial conditions. For simplicity, throughout this entry, all collections of system initial conditions are taken to be independent, zero mean and have finite second moment.

A simplified form of the general case treated in Huang et al. (2007) and Nourian and Caines (2013) is given by the following set of controlled SDEs which for each agent A_i includes state coupling with all other agents:

$$dx_i(t) = \frac{1}{N} \sum_{j=1}^{N} f(t, x_i(t), u_i(t), x_j(t)) dt$$
$$+ \sigma d w_i(t), \quad 1 \leq i \leq N,$$

where here, for the sake of simplicity, only the uniform (non-parameterized) generic agent case is presented. The dynamics of a generic agent in the infinite population limit of this system is then described by the following controlled MV stochastic differential equation:

$$dx(t) = f[t, x(t), u(t), \mu_t] dt + \sigma d w(t),$$

where $f[t, x, u, \mu_t] = \int_{\mathbb{R}^n} f(t, x, u, y) \mu_t(dy)$, with the initial condition measure μ_0 specified, where $\mu_t(\cdot)$ denotes the state distribution of the population at $t \in [0, T]$. The dynamics used in the analysis in Lasry and Lions (2006a,b, 2007) and Cardaliaguet (2012) are of the form $dx_i(t) = u_i(t) dt + \sigma d w_i(t)$.

The dynamical evolution of the state x_i of the ith agent A_i in the discrete time Markov Decision Processes (MDP)-based formulation of the so-called anonymous sequential games (Bergin and Bernhardt 1992; Jovanovic and Rosenthal 1988; Weintraub et al. 2005) is described by a Markov state transition function, or kernel, of the form $P_{t+1} := P(x_i(t+1)|x_i(t), x_{-i}(t), u_i(t), P_t)$.

Agent Performance Functions

In the basic finite population linear-quadratic diffusion case, the agent $A_i, 1 \leq i \leq N$, possesses a performance, or loss, function of the form

$$J_i^N(u_i, u_{-i}) = E \int_0^T \{\|x_i(t) - m_N(t)\|_Q^2 + \|u_i(t)\|_R^2\} dt,$$

where we assume the cost coupling to be of the form $m_N(t) := (\overline{x_N(t)} + \eta)$, $\eta \in \mathbb{R}^n$, where u_{-i} denotes all agents' control laws except for that of the ith agent and $\overline{x_N}$ denotes the population average state $(1/N) \sum_{i=1}^{N} x_i$, and where here and below the expectation is taken over an underlying sample space which carries all initial conditions and Wiener processes.

For the nonlinear case introduced in the previous section, a corresponding finite population mean field loss function is

$$J_i^N(u_i; u_{-i}) :=$$
$$E \int_0^T \left((1/N) \sum_{j=1}^{N} L(t, x_i(t), u_i(t), x_j(t)) \right) dt,$$
$$1 \leq i \leq N,$$

where L is the nonlinear state cost-coupling function. Setting, by possible abuse of notation, $L[t, x, u, \mu_t] = \int_{\mathbb{R}^n} L(t, x, u, y) \mu_t(dy)$, the infinite population limit of this cost function for a generic individual agent A is given by

$$J(u, \mu) := E \int_0^T L[t, x(t), u(t), \mu_t] dt,$$

which is the general expression for the infinite population individual performance functions appearing in Huang et al. (2006) and Nourian and Caines (2013) and which includes those of Lasry and Lions (2006a,b, 2007) and Cardaliaguet (2012). Exponentially discounted costs with discount rate parameter ρ are employed for infinite time horizon performance functions in Huang et al. (2003, 2007), while the sample path

limit of the long-range average is used for ergodic MFG problems in Lasry and Lions (2006a, 2007) and Li and Zhang (2008) and in the analysis of adaptive MFG systems (Kizilkale and Caines 2013).

The Existence of Equilibria

The objective of each agent is to find strategies (i.e., control laws) which are admissible with respect to information and other constraints and which minimize its performance function. The resulting problem is necessarily game theoretic and consequently central results of the topic concern the existence of Nash Equilibria and their properties.

The basic linear-quadratic mean field problem has an explicit solution characterizing a Nash equilibrium (see Huang et al. 2003, 2007). Consider the scalar infinite time horizon discounted case, with nonuniform parameterized agents A_θ with parameter distribution $F(\theta), \theta \in \mathcal{A}$, and dynamical parameters identified as $a_\theta := F_\theta, b_\theta := G_\theta, Q := 1, r := R$; then the so-called Nash Certainty Equivalence (NCE) equation scheme generating the equilibrium solution takes the form

$$\rho s_\theta = \frac{ds_\theta}{dt} + a_\theta s_\theta - \frac{b_\theta^2}{r}\Pi_\theta s_\theta - x^*,$$

$$\frac{d\overline{x}_\theta}{dt} = \left(a_\theta - \frac{b_\theta^2}{r}\Pi_\theta\right)\overline{x}_\theta - \frac{b_\theta^2}{r}s_\theta, \qquad 0 \leq t < \infty,$$

$$\overline{x}(t) = \int_{\mathcal{A}} \overline{x}_\theta(t)dF(\theta),$$

$$x^*(t) = \gamma(\overline{x}(t) + \eta),$$

$$\rho\Pi_\theta = 2a_\theta\Pi_\theta - \frac{b_\theta^2}{r}\Pi_\theta + 1, \quad \Pi_\theta > 0, \qquad \text{Riccati Equation}$$

where the control action of the generic parameterized agent A_θ is given by $u_\theta^0(t) = -\frac{b_\theta}{r}(\Pi_\theta x_\theta(t) + s_\theta(t)), 0 \leq t < \infty$. u_θ^0 is the optimal tracking feedback law with respect to $x^*(t)$ which is an affine function of the mean field term $\overline{x}(t)$, the mean with respect to the parameter distribution F of the $\theta \in \mathcal{A}$ parameterized state means of the agents. Subject to the conditions for the NCE scheme to have a solution, each agent is necessarily in a Nash equilibrium in all full information causal (i.e., non-anticipative) feedback laws with respect to the remainder of the agents when these are employing the law u^0.

It is an important feature of the best response control law u_θ^0 that its form depends only on the parametric data of the entire set of agents, and at any instant it is a feedback function of only the state of the agent A_θ itself and the deterministic mean field-dependent offset s_θ.

For the general nonlinear case, the MFG equations on $[0, T]$ are given by the linked equations for (i) the performance function V for each agent in the continuum, (ii) the FPK for the MV-SDE for that agent, and (iii) the specification of the best response feedback law depending on the mean field measure μ_t and the agent's state $x(t)$. In the uniform agents case, these take the following form.

The Mean Field Game HJB: (MV) FPK Equations

$$[\text{MV-HJB}] \quad -\frac{\partial V(t,x)}{\partial t} = \inf_{u \in U} \left\{ f[t, x(t), u(t), \mu_t] \frac{\partial V(t,x)}{\partial x} + L[t, x(t), u(t), \mu_t] \right\}$$
$$+ \frac{\sigma^2}{2} \frac{\partial^2 V(t,x)}{\partial x^2},$$
$$V(T, x) = 0, \qquad (t, x) \in [0, T] \times \mathbb{R},$$

$$[\text{MV-FPK}] \quad \frac{\partial \mu(t,x)}{\partial t} = -\frac{\partial \{ f[t, x, u(t), \mu_t] \mu(t,x) \}}{\partial x} + \frac{\sigma^2}{2} \frac{\partial^2 \mu(t,x)}{\partial x^2},$$

$$[\text{MV-BR}] \qquad u(t) = \varphi(t, x(t) | \mu_t), \quad (t, x) \in [0, T] \times \mathbb{R}.$$

The general nonlinear MFG problem is approached by different routes in Huang et al. (2006) and Nourian and Caines (2013), and Lasry and Lions (2006a,b, 2007) and Cardaliaguet (2012), respectively. In the former, the so-called probabilistic method solves the MFG equations directly. Subject to technical conditions, an iterated contraction argument establishes the existence of a solution to the HJB-(MV) FPK equations; the best response control laws are obtained from these MFG equations, and these are necessarily Nash equilibria within all causal feedback laws for the infinite population problem. In Lasry and Lions (2006a, 2007) the MFG equations on the infinite time interval (i.e., the ergodic case) are obtained as the limit of Nash equilibria for increasing finite populations, while in the expository notes of Cardaliaguet (2012) the analytic properties of solutions to the HJB-FPK equations on the finite interval are analyzed using PDE methods including the theory of viscosity solutions.

In Huang et al. (2003, 2006, 2007), Nourian and Caines (2013), and Cardaliaguet (2012), it is shown that subject to technical conditions, the solutions to the HJB-FPK scheme yield ε-Nash solutions for finite population MFGs in that for any $\varepsilon > 0$, there exists a population size N_ε such that for all larger populations the use of the feedback law given by the MFG infinite population scheme gives each agent a value to its performance function within ε of the infinite population Nash value.

A counterintuitive feature of these results is that, asymptotically in population size, observations of the states of rival agents are of no value to any given agent; this is in contrast to the situation in single-agent optimal control theory where the value of observations on an agent's environment is in general positive.

Current Developments and Open Problems

There is now an extensive literature on Mean Field Games, the following being a sample: the mathematical literature has focused on the study of general classes of solutions to the fundamental HJB-FPK equations (see e.g., Cardaliaguet (2013)), while in systems and control, the theory of major-minor agent MFG problems (in economics terminology, atoms and continua) is being developed (Huang 2010; Nourian and Caines 2013; Nguyen and Huang 2012), adaptive control extensions of the LQG theory have been carried out (Kizilkale and Caines 2013), and the risk-sensitive case has

been analyzed (Tembine et al. 2012). Much work is now under way in the applications of MFG theory to economics, finance, distributed energy systems, and electrical power markets. Each of these areas has significant open problems, including the application of mathematical transport theory to HJB-FPK equations, the role of MFG theory in portfolio optimization, and the analysis of systems where the presence of partially observed major and minor agent states incurs mean field and agent state estimation.

Cross-References

► Dynamic Noncooperative Games
► Game Theory: Historical Overview
► Stochastic Dynamic Programming
► Stochastic Linear-Quadratic Control
► Stochastic Maximum Principle

Bibliography

Altman E, Basar T, Srikant R (2002) Nash equilibria for combined flow control and routing in networks: asymptotic behavior for a large number of users. IEEE Trans Autom Control 47(6):917–930. Special issue on Control Issues in Telecommunication Networks

Aumann RJ, Shapley LS (1974) Values of non-atomic games. Princeton University Press, Princeton

Basar T, Ho YC (1974) Informational properties of the Nash solutions of two stochastic nonzero-sum games. J Econ Theory 7:370–387

Basar T, Olsder GJ (1999) Dynamic noncooperative game theory. SIAM, Philadelphia

Bensoussan A, Frehse J (1984) Nonlinear elliptic systems in stochastic game theory. J Reine Angew Math 350:23–67

Bergin J, Bernhardt D (1992) Anonymous sequential games with aggregate uncertainty. J Math Econ 21:543–562. North-Holland

Cardaliaguet P (2012) Notes on mean field games. Collège de France

Cardaliaguet P (2013) Long term average of first order mean field games and work KAM theory. Dyn Games Appl 3:473–488

Correa JR, Stier-Moses NE (2010) In: Cochran JJ (ed) Wardrop equilibria. Wiley encyclopedia of operations research and management science. Jhon Wiley & Sons, Chichester, UK

Haurie A, Marcotte P (1985) On the relationship between Nash-Cournot and Wardrop equilibria. Networks 15(3):295–308

Ho YC (1980) Team decision theory and information structures. Proc IEEE 68(6):15–22

Huang MY (2010) Large-population LQG games involving a major player: the Nash certainty equivalence principle. SIAM J Control Optim 48(5):3318–3353

Huang MY, Caines PE, Malhamé RP (2003) Individual and mass behaviour in large population stochastic wireless power control problems: centralized and Nash equilibrium solutions. In: IEEE conference on decision and control, Maui, pp 98–103

Huang MY, Malhamé RP, Caines PE (2006) Large population stochastic dynamic games: closed loop Kean-Vlasov systems and the Nash certainty equivalence principle. Commun Inf Syst 6(3):221–252

Huang MY, Caines PE, Malhamé RP (2007) Large population cost-coupled LQG problems with non-uniform agents: individual-mass behaviour and decentralized ε – Nash equilibria. IEEE Tans Autom Control 52(9):1560–1571

Jovanovic B, Rosenthal RW (1988) Anonymous sequential games. J Math Econ 17(1):77–87. Elsevier

Kizilkale AC, Caines PE (2013) Mean field stochastic adaptive control. IEEE Trans Autom Control 58(4):905–920

Lasry JM, Lions PL (2006a) Jeux à champ moyen. I – Le cas stationnaire. Comptes Rendus Math 343(9):619–625

Lasry JM, Lions PL (2006b) Jeux à champ moyen. II – Horizon fini et controle optimal. Comptes Rendus Math 343(10):679–684

Lasry JM Lions PL (2007) Mean field games. Jpn J Math 2:229–260

Li T, Zhang JF (2008) Asymptotically optimal decentralized control for large population stochastic multiagent systems. IEEE Tans Autom Control 53(7):1643–1660

Neyman A (2002) Values of games with infinitely many players. In: Aumann RJ, Hart S (eds) Handbook of game theory, vol 3. North Holland, Amsterdam, pp 2121–2167

Nourian M, Caines PE (2013) ε-Nash Mean field games theory for nonlinear stochastic dynamical systems with major and minor agents. SIAM J Control Optim 50(5):2907–2937

Nguyen SL, Huang M (2012) Linear-quadratic-Gaussian mixed games with continuum-parametrized minor players. SIAM J Control Optim 50(5):2907–2937

Tembine H, Zhu Q, Basar T (2012) Risk-sensitive mean field games. arXiv:1210.2806

Wardrop JG (1952) Some theoretical aspects of road traffic research. In: Proceedings of the institute of civil engineers, London, part II, vol 1, pp 325–378

Weintraub GY, Benkard C, Van Roy B (2005) Oblivious equilibrium: a mean field approximation for large-scale dynamic games. In: Advances in neural information processing systems. MIT, Cambridge

M

Mechanism Design

Ramesh Johari
Stanford University, Stanford, CA, USA

Abstract

Mechanism design is concerned with the design of strategic environments to achieve desired outcomes at equilibria of the resulting game. We briefly overview central ideas in mechanism design. We survey both objectives the mechanism designer may seek to achieve, as well as equilibrium concepts the designer may use to model agents. We conclude by discussing a seminal example of mechanism design at work: the Vickrey-Clarke-Groves (VCG) mechanisms.

Keywords

Introduction

Informally, *mechanism design* might be considered "inverse game theory." In mechanism design, a principal (the "designer") creates a system (the "mechanism") in which strategic agents interact with each other. Typically, the goal of the mechanism designer is to ensure that at an "equilibrium" of the resulting strategic interaction, a "desirable" outcome is achieved. Examples of mechanism design at work include the following:

1. The FCC chooses to auction spectrum among multiple competing, strategic bidders to maximize the revenue generated. How should the FCC design the auction?
2. A search engine decides to run a market for sponsored search advertising. How should the market be designed?
3. The local highway authority decides to charge tolls for certain roads to reduce congestion. How should the tolls be chosen?

In each case, the mechanism designer is shaping the incentives of participants in the system. The mechanism designer must first define the desired objective and then choose a mechanism that optimizes that objective given a prediction of how strategic agents will respond. The theory of mechanism design provides guidance in solving such optimization problems.

We provide a brief overview of some central concepts in mechanism design. In the first section, we delve into more detail on the structure of the optimization problem that a mechanism designer solves. In particular, we discuss two central features of this problem: (1) What is the objective that the mechanism designer seeks to achieve or optimize? (2) How does the mechanism designer model the agents, i.e., what equilibrium concept describes their strategic interactions? In the second section, we study a specific celebrated class of mechanisms, the Vickrey-Clarke-Groves mechanisms.

Objectives and Equilibria

A mechanism design problem requires two essential inputs, as described in the introduction. First, what is the objective the mechanism designer is trying to achieve or optimize? And second, what are the constraints within which the mechanism designer operates? On the latter question, perhaps the biggest "constraint" in mechanism design is that the agents are assumed to act rationally in response to whatever mechanism is imposed on them. In other words, the mechanism designer needs to model *how* the agents will interact with each other. Mathematically, this is modeled by a choice of equilibrium concept. For simplicity, we focus only on *static* mechanism design, i.e., mechanism design for settings where all agents act simultaneously.

Objectives

In this section we briefly discuss three objectives the mechanism designer may choose to optimize for: *efficiency*, *revenue*, and a *fairness* criterion.

1. **Efficiency**. When the mechanism designer focuses on "efficiency," they are interested in ensuring that the equilibrium outcome of the game they create is a Pareto efficient outcome. In other words, at an equilibrium of the game, there should be no individual that can be made strictly better off while leaving all others at least as well off as they were before. The most important instantiation of the efficiency criterion arises in *quasilinear* settings, i.e., settings where the utility of all agents is measured in a common, transferable monetary unit. In this case, it can be shown that achieving efficient outcomes is equivalent to maximizing the aggregate utility of all agents in the system. See Chap. 23 in Mas-Colell et al. (1995) for more details on mechanism design for efficient outcomes.

2. **Revenue**. Efficiency may be a reasonable goal for an impartial social planner; on the other hand, in many applied settings, the mechanism designer is often herself a profit-maximizing party. In these cases, it is commonly the goal of the mechanism designer to maximize her own payoff from the mechanism itself.

    A common example of this scenario is in the design of *optimal auctions*. An auction is a mechanism for the sale of a good (or multiple goods) among many competing buyers. When the principal is self-interested, she may wish to choose the auction design that maximizes her revenue from sale; the celebrated paper of Myerson (1981) studies this problem in detail.

3. **Fairness**. Finally, in many settings, the mechanism designer may be interested more in achieving a "fair" outcome – even if such an outcome is potentially not Pareto efficient. Fairness is subjective, and therefore, there are many potential objectives that might be viewed as fair by the mechanism designer. One common setting where the mechanism design strives for fair outcomes is in *cost sharing*: in a canonical example, the cost of a project must be shared "fairly" among many participants. See Chap. 15 of Nisan et al. (2007) for more discussion of such mechanisms.

Equilibrium Concepts

In this section we briefly discuss a range of equilibrium concepts the mechanism designer might use to model the behavior of players. From an optimization viewpoint, mechanism design should be viewed as maximization of the designer's objective, subject to an equilibrium constraint. The equilibrium concept used captures the mechanism designer's judgment about how the agents can be expected to interact with each other, once the mechanism designer has fixed the mechanism. Here we briefly discuss three possible equilibrium concepts that might be used by the mechanism designer.

1. **Dominant strategies**. In dominant strategy implementation, the mechanism designer assumes that agents will play a (weak or strict) dominant strategy against their competitors. This equilibrium concept is obviously quite strong, as dominant strategies may not exist in general. However, the advantage is that when the mechanism possesses dominant strategies for each player, the prediction of play is quite strong. The Vickrey-Clarke-Groves mechanisms described below are central in the theory of mechanism design with dominant strategies.

2. **Bayesian equilibrium**. In a Bayesian equilibrium, agents optimize given a common prior distribution about the other agents' preferences. In Bayesian mechanism design, the mechanism designer chooses a mechanism taking into account that the agents will play according to a Bayesian equilibrium of the resulting game. This solution concept allows the designer to capture a lack of complete information among players, but typically allows for a richer family of mechanisms than mechanism design with dominant strategies. Myerson's work on optimal auction design is carried out in a Bayesian framework (Myerson 1981).

3. **Nash equilibrium**. Finally, in a setting where the mechanism designer believes the agents will be quite knowledgeable about each other's preferences, it may be reasonable to assume they will play a Nash equilibrium of the resulting game. Note that in this case,

it is typically assumed the designer does not know the utilities of agents at the time the mechanism is chosen – even though agents *do* know their own utilities at the time the resulting game is actually played. See, e.g., Moore (1992) for an overview of mechanism design with Nash equilibrium as the solution concept.

The Vickrey-Clarke-Groves Mechanisms

In this section, we describe a seminal example of mechanism design at work: the *Vickrey-Clarke-Groves* (VCG) class of mechanisms. The key insight behind VCG mechanisms is that by structuring payment rules correctly, individuals can be incentivized to truthfully declare their utility functions to the market and in turn achieve an efficient allocation. VCG mechanisms are an example of mechanism design with dominant strategies and with the goal of welfare maximization, i.e., efficiency. The presentation here is based on the material in Chap. 5 of Berry and Johari (2011), and the reader is referred there for further discussion. See also Vickrey (1961), Clarke (1971), and Groves (1973) for the original papers discussing this class of mechanisms.

To illustrate the principle behind VCG mechanisms, consider a simple example where we allocate a single resource of unit capacity among R competing users. Each user's utility is measured in terms of a common currency unit; in particular, if the allocated amount is x_r and the payment to user r is t_r, then her utility is $U_r(x_r) + t_r$; we refer to U_r as the *valuation* function, and let the space of valuation functions be denoted by $\mathcal{U}$. For simplicity we assume the valuation functions are continuous. In line with our discussion of efficiency above, it can be shown that the Pareto efficient allocation is obtained by solving the following:

$$\text{maximize} \quad \sum_r U_r(x_r) \qquad (1)$$

$$\text{subject to} \quad \sum_r x_r \leq 1; \qquad (2)$$

$$\mathbf{x} \geq 0. \qquad (3)$$

However, achieving the efficient allocation requires knowledge of the utility functions; what can we do if these are unknown? The key insight is to make each user act *as if* they are optimizing the aggregate utility, by structuring payments appropriately. The basic approach in a VCG mechanism is to let the strategy space of each user r be the set $\mathcal{U}$ of possible valuation functions and make a payment t_r to user r so that her net payoff has the same form as the social objective (1). In particular, note that if user r receives an allocation x_r and a payment t_r, the payoff to user r is

$$U_r(x_r) + t_r.$$

On the other hand, the social objective (1) can be written as

$$U_r(x_r) + \sum_{s \neq r} U_s(x_s).$$

Comparing the preceding two expressions, the most natural means to align user objectives with the social planner's objectives is to *define the payment t_r as the sum of the valuations of all users other than r.*

A VCG mechanism first asks each user to declare a valuation function. For each r, we use $\hat{U}_r$ to denote the declared valuation function of user r and use $\hat{\mathbf{U}} = (\hat{U}_1, \ldots, \hat{U}_R)$ to denote the vector of declared valuations. Formally, given a vector of declared valuation functions $\hat{\mathbf{U}}$, a VCG mechanism chooses the allocation $\mathbf{x}(\hat{\mathbf{U}})$ as an optimal solution to (1)–(2) given $\hat{\mathbf{U}}$, i.e.,

$$\mathbf{x}(\hat{\mathbf{U}}) \in \arg \max_{\mathbf{x} \geq 0 : \sum_r x_r \leq 1} \sum_r \hat{U}_r(x_r). \qquad (4)$$

The payments are then structured so that

$$t_r(\hat{\mathbf{U}}) = \sum_{s \neq r} \hat{U}_s(x_s(\hat{\mathbf{U}})) + h_r(\hat{\mathbf{U}}_{-r}). \qquad (5)$$

Here h_r is an arbitrary function of the declared valuation functions of users other than r, and various definitions of h_r give rise to variants of the VCG mechanisms. Since user r cannot affect this term through the choice of $\hat{U}_r$, she chooses $\hat{U}_r$ to maximize

$$U_r(x_r(\hat{\mathbf{U}})) + \sum_{s \neq r} \hat{U}_s(x_s(\hat{\mathbf{U}})).$$

Now note that given $\hat{\mathbf{U}}_{-r}$, the above expression is bounded above by

$$\max_{\mathbf{x} \geq 0: \sum_r x_r \leq 1} \left[U_r(x_r) + \sum_{s \neq r} \hat{U}_s(x_s) \right].$$

But since $\mathbf{x}(\hat{\mathbf{U}})$ satisfies (4), user r can achieve the preceding maximum by truthfully declaring $\hat{U}_r = U_r$. Since this optimal strategy does not depend on the valuation functions $(\hat{U}_s, s \neq r)$ declared by the other users, we recover the important fact that in a VCG mechanism, *truthful declaration is a weak dominant strategy for user r*.

For our purposes, the interesting feature of the VCG mechanism is that it elicits the true utilities from the users and in turn (because of the definition of $\mathbf{x}(\hat{\mathbf{U}})$) chooses an efficient allocation. The feature that truthfulness is a dominant strategy is known as *incentive compatibility*: the individual incentives of users are aligned, or "compatible," with overall efficiency of the system. The VCG mechanism achieves this by effectively paying each agent to tell the truth. The significance of the approach is that this payment can be properly structured even if the resource manager does not have prior knowledge of the true valuation functions.

Summary and Future Directions

Mechanism design provides an overarching framework for the "engineering" of economic systems. However, many significant challenges remain. First, VCG mechanisms are not computationally tractable in complex settings, e.g., combinatorial auctions (Hajek 2013); finding computationally tractable yet efficient mechanisms is a very active area of current research. Second, VCG mechanisms optimize for overall welfare, rather than revenue, and finding simple mechanisms that maximize revenue also presents new challenges. Finally,

we have only considered implementation in static environments. Most practical mechanism design settings are dynamic. Dynamic mechanism design remains an active area of fruitful research.

Cross-References

▶ Auctions
▶ Game Theory: Historical Overview
▶ Linear Quadratic Zero-Sum Two-Person Differential Games

Bibliography

Berry RA, Johari R (2011) Economic modeling in networking: a primer. Found Trends Netw 6(3):165–286
Clarke EH (1971) Multipart pricing of public goods. Public Choice 11(1):17–33
Groves T (1973) Incentives in teams. Econometrica 41(4):617–631
Hajek B (2013) Auction theory. In: Encyclopedia of systems and control. Springer
Mas-Colell A, Whinston M, Green J (1995) Microeconomic theory. Oxford University Press, New York
Moore J (1992) Implementation, contracts, and renegotiation in environments with complete information. Adv Econ Theory 1:182–281
Myerson RB (1981) Optimal auction design. Math Oper Res 6(1):58–73
Nisan N, Roughgarden T, Tardos E, Vazirani V (eds) (2007) Algorithmic game theory. Cambridge University Press, Cambridge/New York
Vickrey W (1961) Counterspeculation, auctions, and competitive sealed tenders. J Financ 16(1): 8–37

M

Model Building for Control System Synthesis

Marco Lovera and Francesco Casella
Politecnico di Milano, Milan, Italy

Abstract

The process of developing control-oriented mathematical models of physical systems is a complex task, which in general implies a careful combination of prior knowledge about the physics of

the system under study with information coming from experimental data. In this article the role of mathematical models in control system design and the problem of developing compact control-oriented models are discussed.

Keywords

Analytical models; Computational modeling; Continuous-time systems; Control-oriented modeling; Discrete-time systems; Parameter-varying systems; Simulation; System identification; Time-invariant systems; Time-varying systems; Uncertainty

Introduction

The design of automatic control systems requires the availability of some knowledge of the dynamics of the process to be controlled. In this respect, current methods for control system synthesis can be classified in two broad categories: model-free and model-based ones.

The former aim at designing (or tuning) controllers solely on the basis of experimental data collected directly on the plant, without resorting to mathematical models.

The latter, on the contrary, assume that suitable models of the plant to be controlled are available, and rely on this information to work out control laws capable of meeting the design requirements.

While the research on model-free design methods is a very active field, the vast majority of control synthesis methods and tools fall in the model-based category and therefore assume that knowledge about the plant to be controlled is encoded in the form of dynamic models of the plant itself. Furthermore, in an increasing number of application areas, control system performance is becoming a key competitive factor for the success of innovative, high-tech systems. Consider, for example, high-performance mechatronic systems (such as robots); vehicles enhanced by active integrated stability, suspension, and braking control;

aerospace systems; advanced energy conversion systems. All the abovementioned applications possess at least one of the following features, which in turn call for accurate mathematical modeling for the design of the control system: closed-loop performance critically depends on the dynamic behavior of the plant; the system is made of many closely interacting subsystems; advanced control systems are required to obtain competitive performance, and these in turn depend on explicit mathematical models for their design; the system is safety critical and requires extensive validation of closed-loop stability and performance by simulation.

Therefore, building control-oriented mathematical models of physical systems is a crucial prerequisite to the design process itself (see, e.g., Lovera (2014) for a more detailed treatment of this topic).

In the following, two aspects related to modeling for control system synthesis will be discussed, namely, the role of models for control system synthesis and the actual process of model building itself.

The Role of Models for Control System Synthesis

Mathematical models play a number of different roles in the design of control systems. In particular, different classes of mathematical models are usually employed: *detailed*, high-fidelity models for system simulation and *compact* models for control design. In this section the two model classes are presented and their respective roles in the design of control systems are described. Note, in passing, that although hybrid system control is an interesting and emerging field, this entry will focus on purely continuous-time physical models, with application to the design of continuous-time or sampled-time control systems.

Detailed Models for System Simulation
Detailed models play a double role in the control design process. On one hand, they allow checking how good (or crude) the compact model is, compared to a more detailed description, thus helping

to develop good compact models. On the other hand, they allow closed-loop performance verification of the controlled system, once a controller design is available. Indeed, verifying closed-loop performance using the same simplified model that was used for control system design is not a sound practice; conversely, verification performed with a more detailed model is usually deemed a good indicator of the control system performance, whenever experimental validation is not possible for some reason.

Object-oriented modeling (OOM) methodologies and equation-based, object-oriented languages (EOOLs) provide very good support for the development of such high-fidelity models, thanks to equation-based modeling, acausal physical ports, hierarchical system composition, and inheritance; see Tiller (2001) for a comprehensive overview. Any continuous-time EOOL model is equivalent to the set of differential-algebraic equations (DAEs)

$$F(x(t), \dot{x}(t), u(t), y(t), p, t) = 0, \quad (1)$$

where x is the vector of dynamic variables, u is the vector of input variables, y is the vector of algebraic variables, p is the vector of parameters and t is the time. It is interesting to highlight that the object-oriented approach (in particular, the use of replaceable components) allows defining and managing families of models of the same plant with different levels of complexity, by providing more or less detailed implementations of the same abstract interfaces. This feature of OOM allows the development of simulation models for different purposes and with different degrees of detail throughout the entire life of an engineering project, from preliminary design down to commissioning and personnel training, all within a coherent framework.

In particular, when focusing on control systems verification (and regardless of the actual control design methodology) once the controller has been set up, an OOM tool can be used to run closed-loop simulations, including both the plant and the controller model. Many OOM tools provide model export facilities, which allow to connect a plant model with only causal external connectors (actuator inputs and sensor outputs) to a causal controller model in a causal simulation environment. From a mathematical point of view, this corresponds to reformulating (1) in state-space form, by means of analytical and/or numerical transformations.

Finally, it is important to point out that physical model descriptions based on partial-differential equations (PDEs) can be handled in the OOM framework by means of discretization using finite volume, finite elements, or finite differences methods.

Compact Models for Control Design

The requirements for a control-oriented model can vary significantly from application to application. Design models can be tentatively classified in terms of two key features: complexity and accuracy. For a dynamic model, complexity can be measured in terms of its order; accuracy, on the other hand, can be measured using many different metrics (e.g., time-domain simulation or prediction error, frequency domain matching with the real plant, etc.) related to the capability of the model to reproduce the behavior of the true system in the operating conditions of interest.

Broadly speaking, it can be safely stated that the required level of closed-loop performance drives the requirements on the accuracy and complexity of the design model. Similarly, it is intuitive that more complex models have the potential for being more accurate. So, one might be tempted to resort to very detailed mathematical representations of the plant to be controlled in order to maximize closed-loop performance. This consideration however is moderated by a number of additional requirements, which actually end up driving the control-oriented modeling process. First of all, present-day controller synthesis methods and tools have computational limitations in terms of the complexity of the mathematical models they can handle, so compact models representative of the dominant dynamics of the system under study are what is really needed. Furthermore, for many synthesis methods (such as, e.g., LQG or H_∞ synthesis), the complexity of the design model has an impact on the complexity of the controller,

which in turn is constrained by implementation issues. Last but not least, in engineering projects, the budget of the control-oriented modeling activity is usually quite limited, so the achievable level of accuracy is affected by this limitation.

It is clear from the above discussion that developing mathematical models suitable for control system synthesis is a nontrivial task but rather corresponds to the pursuit of a careful tradeoff between complexity and accuracy. Furthermore, throughout the model development, one should keep in mind the eventual control application of the model, so its mathematical structure has to be compatible with currently available methods and tools for control system analysis and design.

Control-oriented models are usually formulated in state-space form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), p, t) \\ y(t) &= g(x(t), u(t), p, t) \end{aligned} \quad (2)$$

where x is the vector of state variables, u is the vector of system inputs (control variables and disturbances), y is the vector of system outputs, p is the vector of parameters, and t is the continuous time. In the following, however, the focus will be on linear models, which constitute the starting point for most control law design methods and tools. In this respect, the main categories of models used in control system synthesis can be defined as follows.

Linear Time-Invariant Models

Linear time-invariant (LTI) models can be described in state-space form as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (3)$$

or, equivalently, using an input-output model given by the (rational) transfer function

$$G(s) = C(sI - A)^{-1}B + D, \quad (4)$$

where s denotes the Laplace variable. In many cases, the dynamics of systems in the form (2) in the neighborhood of an equilibrium (trim) point

is approximated by (3) via analytical or numerical linearization.

If, on the contrary, the control-oriented model is obtained by linearization of the DAE system (1), then a generalized LTI (or descriptor) model in the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (5)$$

is obtained. Clearly, a generalized LTI model is equivalent to a conventional one as long as E is nonsingular. The generalized form, however, encompasses the wider class of linearized plants with a singular E.

Linear Time-Varying Models

In some engineering applications, the need may arise to linearize the detailed model in the neighborhood of a trajectory rather than around an equilibrium point. Whenever this is the case, a linear time-varying (LTV) model is obtained, in the form

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t). \end{aligned} \quad (6)$$

An important subclass is the one of time periodic behavior of the state-space matrices of the model, which corresponds to a linear time periodic (LTP) model. LTP models arise when considering the linearization along periodic trajectories or, as it occurs in a number of engineering problems, whenever rotating systems are considered (e.g., spacecraft, rotorcraft, wind turbines). Finally, it is interesting to recall that (discrete-time) LTP models are needed to model multi-rate sampled data systems.

Linear Parameter-Varying models

The control-oriented modeling problem can be also formulated as the one of *simultaneously* representing all the linearizations of interest for control purposes of a given nonlinear plant. Indeed, in many control engineering applications a single control system must be designed to guarantee the satisfactory closed-loop operation of a given plant in many different operating

conditions (either equilibria or trajectories). Many design techniques are now available for this problem (see, e.g., Mohammadpour and Scherer 2012), provided that a suitable model in parameter-dependent form has been derived for the system to be controlled. Linear parameter-varying (LPV) models, described in state-space form as

$$\begin{aligned} \dot{x}(t) &= A(p(t))x(t) + B(p(t))u(t) \\ y(t) &= C(p(t))x(t) + D(p(t))u(t) \end{aligned} \quad (7)$$

are linear models the state-space representation of which depends on a parameter vector p that can be time varying. The elements of vector p may or may not be measurable, depending on the specific problem formulation. The present state of the art of LPV modeling can be briefly summarized by defining two classes of approaches (see Lopes dos Santos et al. (2011) for details). *Analytical* methods based on the availability of reliable nonlinear equations for the dynamics of the plant, from which suitable control-oriented representations can be derived (by resorting to, broadly speaking, suitable extensions of the familiar notion of linearization, developed in order to take into account off-equilibrium operation of the system). *Experimental* methods based entirely on identification, i.e., aimed at deriving LPV models for the plant directly from input/output data. In particular, some LPV identification techniques assume that one *global* identification experiment in which both the control input and the parameter vector are (persistently) excited in a simultaneous way, while others aim at deriving a parameter-dependent model on the basis of *local* experiments only, i.e., experiments in which the parameter vector is held constant and only the control input is excited.

Modern control theory provides methods and tools to deal with design problems in which stability and performance have to be guaranteed also in the presence of model uncertainty, both for regulation around a specified operating point and for gain scheduled control system design. Therefore, modeling for control system synthesis should also provide methods to account for model uncertainty (both parametric and nonparametric) in the considered model class.

Most of the existing control design literature assumes that the plant model is given in the form of a linear fractional transformation (LFT) (see, e.g., Skogestad and Postlethwaite (2007) for an introduction to LFT modeling of uncertainty and Hecker et al. (2005) for a discussion of algorithms and software tools). LFT models consist of a feedback interconnection between a *nominal* LTI plant and a (usually norm-bounded) operator which represents model uncertainties, e.g., poorly known or time-varying parameters, nonlinearities, etc. A generic such LFT interconnection is depicted in Fig. 1, where the nominal plant is denoted with P and the uncertainty block is denoted with Δ. The LFT formalism can be also used to provide a structured representation for the state-space form of LPV models, as depicted in Fig. 2, where the block $\Delta(\alpha)$ takes into account the presence of the uncertain parameter vector α, while the block $\Delta(p)$ models the effect of the varying operating point, parameterized by the vector of time-varying parameters p. Therefore, LFT models can be used for the design of robust and gain scheduling controllers; in addition they can also serve as a basis for structured model identification techniques, where the uncertain parameters that appear in the feedback blocks are estimated based on input/output data sequences. The process of extracting uncertain/scheduling parameters from the design model of the system
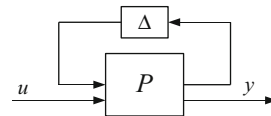
Model Building for Control System Synthesis, Fig. 1 Block diagram of the typical LFT interconnection adopted in the robust control framework
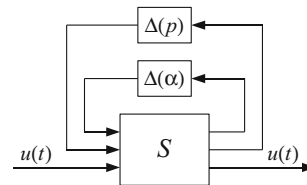
Model Building for Control System Synthesis, Fig. 2 Block diagram of the typical LFT interconnection adopted in the robust LPV control framework

to be controlled is a highly complex one, in which symbolic techniques play a very important role. Tools already exist to perform this task (see, e.g., Hecker et al. 2005), while a recent overview of the state of the art in this research area can be found in Hecker and Varga (2006).

Finally, it is important to point out that there is a vast body of advanced control techniques which are based on discrete-time models:

$$x(k + 1) = f(x(k), u(k), p, k)$$
$$y(k) = g(x(k), u(k), p, k) \qquad (8)$$

where the integer time step k usually corresponds to multiples of a sampling period T_s. Many techniques are available to transform (2) into (8). Furthermore, LTI, LTV, and LPV models can be formulated in discrete time rather than in continuous time.

Building Models for Control System Synthesis

The development of control-oriented models of physical systems is a complex task, which in general implies a careful combination of prior knowledge about the physics of the system under study with information coming from experimental data. In particular, this process can follow very different paths depending on the type of information which is available on the plant to be controlled. Such paths are typically classified in the literature as follows (see, e.g., Ljung (2008) for a more detailed discussion).

White box modeling refers to the development of control-oriented models on the basis of first principles only. In this framework, one uses the available information on the plant to develop a detailed model using OOM or EOOL tools and subsequently works out a compact control-oriented model from it. If the adopted tool only supports simulation, then one can run simulations of the plant model, subject to suitably chosen excitation inputs (ranging from steps to persistently exciting input sequences such as, e.g., pseudorandom binary sequences and sine

sweeps) and then reconstruct the dynamics by means of system identification methods. Note that in this way the structure/order selection stage of the system identification process provides effective means to manage the complexity versus accuracy tradeoff in the derivation of the compact model. A more direct approach, presently supported by many tools, is to directly compute the A, B, C, D matrices of the linearized system around specified equilibrium (trim) points, using symbolic and/or numerical linearization techniques. The result is usually a high-order linear system, which then can (sometimes must) be reduced to a low-order system by using model order reduction techniques (such as, e.g., balanced truncation). Model reduction techniques (see Antoulas (2009) for an in-depth treatment of this topic) allow to automatically obtain approximated compact models such as (3), starting from much more detailed simulation models, by formulating specific approximation bounds in control-relevant terms (e.g., percentage errors of steady-state output values, norm-bounded additive or multiplicative errors of weighted transfer functions, or L_2-norm errors of output transients in response to specified input signals).

Black box modeling, on the other hand, corresponds to situations in which the modeling activity is entirely based on input-output data collected on the plant (which therefore must be already available), possibly in dedicated, suitably designed, experiments (see Ljung 1999). Regardless of the type of model to be built (i.e., linear or nonlinear, time invariant or time varying, discrete time or continuous time), the black box approach consists of a number of well-defined steps. First of all the structure of the model to be identified must be defined: in the linear time-invariant case, this corresponds to the choice of the number of poles and zeros for an input-output model or to the choice of model order for a state-space representation; in the nonlinear case structure selection is a much more involved process in view of the much larger number of degrees of freedom which are potentially involved. Once a model structure has been defined, a suitable

cost function to measure the model performance must be selected (e.g., time-domain simulation or prediction error, frequency domain model fitting, etc.) and the experiments to collect identification and validation data must be designed. Finally, the uncertain model parameters must be estimated from the available identification dataset and the model must be validated on the validation dataset.

Grey box modeling (in various shades) corresponds to the many possible intermediate cases which can occur in practice, ranging from the white box approach to the black box one. As recently discussed in Ljung (2008), the critical issue in the development of an effective approach to control-oriented grey box modeling lies in the integration of existing methods and tools for physical systems modeling and simulation with methods and tools for parameter estimation. Such integration can take place in a number of different ways depending on the relative role of data and priors on the physics of the system in the specific application. A typical situation which occurs frequently in applications is when a white box model (developed by means of OOM or EOOL tools) contains parameters having unknown or uncertain numerical values (such as, e.g., damping factors in structural models, aerodynamic coefficients in aircraft models and so on). Then, one may rely on input-output data collected in dedicated experiments on the real system to refine the white box model by estimating the parameters using the information provided by the data. This process is typically dependent on the specific application domain as the type of experiment, the number of measurements, and the estimation technique must meet application-specific constraints (see, e.g., Klein and Morelli (2006) for an overview of grey box modeling in aerospace applications).

Summary and Future Directions

In this article the problem of model building for control system synthesis has been considered. An overview of the different uses of mathematical models in control system design has been provided and the process of building compact control-oriented models starting from prior knowledge about the system and/or experimental data has been discussed. Present-day modeling and simulation tools support advanced control system design in a much more direct way. In particular, while methods and tools for the individual steps in the modeling process (such as OOM, linearization and model reduction, parameter estimation) are available, an integrated environment enabling the pursuit of all the abovementioned paths to the development of compact control-oriented models is still a subject for future development. The availability of such a tool might further promote the application of advanced, model-based techniques that are currently limited by the model development process.

Cross-References

▶ Modeling of Dynamic Systems from First Principles
▶ Multi-domain Modeling and Simulation
▶ System Identification: An Overview

Bibliography

Antoulas A (2009) Approximation of large-scale dynamical systems. SIAM, Philadelphia

Hecker S, Varga A (2006) Symbolic manipulation techniques for low order LFT-based parametric uncertainty modelling. Int J Control 79(11): 1485–1494

Hecker S, Varga A, Magni J-F (2005) Enhanced LFR-toolbox for MATLAB. Aerosp Sci Technol 9(2):173–180

Klein V, Morelli EA (2006) Aircraft system identification: theory and practice. AIAA, Reston

Ljung L (1999) System identification: theory for the user. Prentice-Hall, New Jersey

Ljung L (2008) Perspectives on system identification. In: 2008 IFAC world congress, Seoul

Lopes dos Santos P, Azevedo Perdicoulis TP, Novara C, Ramos JA, Rivera DE (eds) (2011) Linear parameter-varying system identification: new developments and trends. World Scientific, Singapore

Lovera M (ed) (2014) Control-oriented modelling and identification: theory and practice. IET, London

Mohammadpour J, Scherer C (eds) (2012) Control of linear parameter varying systems with applications. Springer, New York

Skogestad S, Postlethwaite I (2007) Multivariable feedback control analysis and design. Wiley, Chichester/New York

Tiller M (2001) Introduction to physical modelling with Modelica. Kluwer, Boston

MHE

▶ Moving Horizon Estimation

Model Order Reduction: Techniques and Tools

Peter Benner[1] and Heike Faßbender[2]
[1]Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany
[2]Institut Computational Mathematics, Technische Universität Braunschweig, Braunschweig, Germany

Abstract

Model order reduction (MOR) is here understood as a computational technique to reduce the order of a dynamical system described by a set of ordinary or differential-algebraic equations (ODEs or DAEs) to facilitate or enable its simulation, the design of a controller, or optimization and design of the physical system modeled. It focuses on representing the map from inputs into the system to its outputs, while its dynamics are treated as a black box so that the large-scale set of describing ODEs/DAEs can be replaced by a much smaller set of ODEs/DAEs without sacrificing the accuracy of the input-to-output behavior.

Keywords

Problem Description

This survey is concerned with linear time-invariant (LTI) systems in state-space form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{1}$$

where $E, A \in \mathbb{R}^{n \times n}$ are the system matrices, $B \in \mathbb{R}^{n \times m}$ is the input matrix, $C \in \mathbb{R}^{p \times n}$ is the output matrix, and $D \in \mathbb{R}^{p \times m}$ is the feedthrough (or input–output) matrix. The size n of the matrix A is often referred to as the order of the LTI system. It mainly determines the amount of time needed to simulate the LTI system.

Such LTI systems often arise from a finite element modeling using commercial software such as ANSYS or NASTRAN which results in a second-order differential equation of the form

$$\begin{aligned} M\ddot{x}(t) + D\dot{x}(t) + Kx(t) &= Fu(t), \\ y(t) &= C_p x(t) + C_v \dot{x}(t), \end{aligned}$$

where the mass matrix M, the stiffness matrix K, and the damping matrix D are square matrices in $\mathbb{R}^{s \times s}$, $F \in \mathbb{R}^{s \times m}$, $C_p, C_v \in \mathbb{R}^{q \times s}$, $x(t) \in \mathbb{R}^s$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^q$. Such second-order differential equations are typically transformed to a mathematically equivalent first-order differential equation

$$\underbrace{\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix}}_{\dot{z}(t)} = \underbrace{\begin{bmatrix} 0 & I \\ -K & -D \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}}_{z(t)} + \underbrace{\begin{bmatrix} 0 \\ F \end{bmatrix}}_{B} u(t)$$

$$y(t) = \underbrace{\begin{bmatrix} C_p & C_v \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}}_{z(t)},$$

where $E, A \in \mathbb{R}^{2s \times 2s}$, $B \in \mathbb{R}^{2s \times m}$, $C \in \mathbb{R}^{q \times 2s}$, $z(t) \in \mathbb{R}^{2s}$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^q$. Various other linearizations have been proposed in the literature.

The matrix E may be singular. In that case the first equation in (1) defines a system of differential-algebraic equations (DAEs); otherwise it is a system of ordinary differential equations (ODEs). For example, for $E = \begin{bmatrix} J & 0 \\ 0 & 0 \end{bmatrix}$ with a $j \times j$ nonsingular matrix J, only the first j equations in the left-hand side expression in (1) form ordinary differential equations, while the last $n - j$ equations form homogeneous linear equations. If further $A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$ and $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$ with the $j \times j$ matrix A_{11}, the $j \times m$ matrix B_1 and a nonsingular matrix A_{22}, this is easily seen: partitioning the state vector $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$ with $x_1(t)$ of length j, the DAE $E\dot{x}(t) = Ax(t) + Bu(t)$ splits into the algebraic equation $0 = A_{22}x_2(t) + B_2u_2(t)$, and the ODE

$$J\dot{x}_1(t) = A_{11}x_1(t) + \left(B_1 - A_{12}A_{22}^{-1}B_2 \right) u(t).$$

To simplify the description, only continuous-time systems are considered here. The discrete-time case can be treated mostly analogously; see, e.g., Antoulas (2005).

An alternative way to represent LTI systems is provided by the transfer function matrix (TFM), a matrix-valued function whose elements are rational functions. Assuming $x(0) = 0$ and taking Laplace transforms in (1) yields $sX(s) = AX(s) + BU(s)$, $Y(s) = CX(s) + DU(s)$, where $X(s)$, $Y(s)$, and $U(s)$ are the Laplace transforms of the time signals $x(t)$, $y(t)$ and $u(t)$, respectively. The map from inputs U to outputs Y is then described by $Y(s) = G(s)U(s)$ with the TFM

$$G(s) = C(sE - A)^{-1}B + D, \qquad s \in \mathbb{C}. \quad (2)$$

The aim of model order reduction is to find an LTI system

$$\widetilde{E}\dot{\tilde{x}}(t) = \widetilde{A}\tilde{x}(t) + \widetilde{B}u(t), \;\; \tilde{y}(t) = \widetilde{C}\tilde{x}(t) + \widetilde{D}u(t) \quad (3)$$

of reduced-order $r \ll n$ such that the corresponding TFM

$$\widetilde{G}(s) = \widetilde{C}(s\widetilde{E} - \widetilde{A})^{-1}\widetilde{B} + \widetilde{D} \quad (4)$$

approximates the original TFM (2). That is, using the same input $u(t)$ in (1) and (3), we want that the output $\tilde{y}(t)$ of the reduced order model (ROM) (3) approximates the output $y(t)$ of (1) well enough for the application considered (e.g., controller design). In general, one requires $\|y(t) - \tilde{y}(t)\| \leq \varepsilon$ for all feasible inputs $u(t)$, for (almost) all t in the time domain of interest, and for a suitable norm $\| \cdot \|$. In control theory one often employs the $\mathcal{L}_2$- or $\mathcal{L}_\infty$-norms on $\mathbb{R}$ or $[0, \infty]$, respectively, to measure time signals or their Laplace transforms. In the situations considered here, the $\mathcal{L}_2$-norms employed in frequency and time domain coincide due to the Paley-Wiener theorem (or Parseval's equation or the Plancherel theorem, respectively); see Antoulas (2005) and Zhou et al. (1996) for details. As $Y(s) - \widetilde{Y}(s) = (G(s) - \widetilde{G}(s))U(s)$, one can therefore consider the approximation error of the TFM $\|G(\cdot) - \widetilde{G}(\cdot)\|$ measured in an induced norm instead of the error in the output $\|y(\cdot) - \tilde{y}(\cdot)\|$.

Depending on the choice of the norm, different MOR goals can be formulated. Typical choices are (see, e.g., Antoulas (2005) for a more thorough discussion)

• $\|G(\cdot) - \widetilde{G}(\cdot)\|_{\mathcal{H}_\infty}$, where

$$\|F(.)\|_{\mathcal{H}_\infty} = \sup_{s \in \mathbb{C}_+} \sigma_{\max}(F(s)).$$

Here, $\sigma_{\max}$ is the largest singular value of the matrix $F(s)$. This minimizes the maximal magnitude of the frequency response of the error system and by the Paley-Wiener theorem bounds the $\mathcal{L}_2$-norm of the output error.

• $\|G(\cdot) - \widetilde{G}(\cdot)\|_{\mathcal{H}_2}$, where (with $\iota = \sqrt{-1}$)

$$\|F(\cdot)\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{tr}\left(F(\iota\omega)^* F(\iota\omega) \right) d\omega.$$

This ensures a small error $\|y(\cdot) - \tilde{y}(\cdot)\|_{\mathcal{L}_\infty(0,\infty)} = \sup_{t>0} \|y(t) - \tilde{y}(t)\|_\infty$ (with $\| . \|_\infty$ denoting the maximum norm of a vector)

uniformly over all inputs $u(t)$ having bounded $\mathcal{L}_2$-energy, that is, $\int_0^\infty u(t)^T u(t) dt \leq 1$; see Gugercin et al. (2008).

Besides a small approximation error, one may impose additional constraints for the ROM. One might require certain properties (such as stability and passivity) of the original systems to be preserved. Rather than considering the full nonnegative real line in time domain or the full imaginary axis in frequency domain, one can also consider bounded intervals in both domains. For these variants, see, e.g., Antoulas (2005) and Obinata and Anderson (2001).

Methods

There are a number of different methods to construct ROMs, see, e.g., Antoulas (2005), Benner et al. (2005), Obinata and Anderson (2001), and Schilders et al. (2008). Here we concentrate on projection-based methods which restrict the full state $x(t)$ to an r-dimensional subspace by choosing $\tilde{x}(t) = W^* x(t)$, where W is an $n \times r$ matrix. Here the conjugate transpose of a complex-valued matrix Z is denoted by Z^*, while the transpose of a matrix Y will be denoted by Y^T. Choosing $V \in \mathbb{C}^{n \times r}$ such that $W^* V = I \in \mathbb{R}^{r \times r}$ yields an $n \times n$ projection matrix $\Pi = V W^*$ which projects onto the r-dimensional subspace spanned by the columns of V along the kernel of W^*. Applying this projection to (1), one obtains the reduced-order LTI system (3) with

$$\widetilde{E} = W^* E V, \ \tilde{A} = W^* A V, \ \widetilde{B} = W^* B, \ \widetilde{C} = C V \tag{5}$$

and an unchanged $\widetilde{D} = D$. If $V = W$, Π is an orthogonal projector and is called a Galerkin projection. If $V \neq W$, Π is an oblique projector, sometimes called a Petrov-Galerkin projection.

In the following, we will briefly discuss the main classes of methods to construct suitable matrices V and W: truncation-based methods and interpolation-based methods. Other methods, in particular combinations of the two classes discussed here, can be found in the literature. In case the original LTI system is real, it is often desirable

to construct a real reduced-order model. All of the methods discussed in the following either do construct a real reduced-order system or there is a variant of the method which does. In order to keep this exposition at a reasonable length, the reader is referred to the cited literature.

Truncation Based Methods

The general idea of truncation is most easily explained by *modal truncation*: For simplicity, assume that $E = I$ and that A is diagonalizable, $T^{-1} A T = D_A = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Further we assume that the eigenvalues $\lambda_\ell \in \mathbb{C}$ of A can be ordered such that

$$\text{Re}(\lambda_n) \leq \text{Re}(\lambda_{n-1}) \leq \ldots \leq \text{Re}(\lambda_1) < 0, \tag{6}$$

(i.e., all eigenvalues lie in the open left half complex plane). This implies that the system is stable. Let V be the $n \times r$ matrix consisting of the first r columns of T and let W^* be the first r rows of T^{-1}, that is, $W = V(V^* V)^{-1}$. Applying the transformation T to the LTI system (1) yields

$$T^{-1} \dot{x}(t) = (T^{-1} A T) T^{-1} x(t) + (T^{-1} B) u(t) \tag{7}$$

$$y(t) = (C T) T^{-1} x(t) + D u(t) \tag{8}$$

with

$$T^{-1} A T = \begin{bmatrix} W^* A V & \\ & A_2 \end{bmatrix}, \ T^{-1} B = \begin{bmatrix} W^* B \\ B_2 \end{bmatrix},$$

and $C T = [C V \ C_2]$, where $W^* A V = \text{diag}(\lambda_1, \ldots, \lambda_r)$ and $A_2 = \text{diag}(\lambda_{r+1}, \ldots, \lambda_n)$. Preserving the r dominant poles (eigenvalues with largest real part) by truncating the rest (i.e., discarding A_2, B_2, and C_2 from (7)) yields the ROM as in (5). It can be shown that the error bound

$$\|G(\cdot) - \widetilde{G}(\cdot)\|_{\mathcal{H}_\infty} \leq \|C_2\| \ \|B_2\| \frac{1}{|\text{Re}(\lambda_{r+1})|}$$

holds (Benner 2006). As eigenvalues contain only limited information about the system, this is not necessarily a meaningful reduced-order system. In particular, the dependence of the input–output relation on B and C is completely ignored.

This can be enhanced by more refined dominance measures taking B and C into account; see, e.g., Varga (1995) and Benner et al. (2011).

More suitable reduced-order systems can be obtained by *balanced truncation*. To introduce this concept, we no longer need to assume A to be diagonalizable, but we require the stability of A in the sense of (6). For simplicity, we assume $E = I$. For treatment of the DAE case ($E \neq I$), see Benner et al. (2005, Chap. 3). Loosely speaking, a balanced representation of an LTI system is obtained by a change of coordinates such that the states which are hard to reach are at the same time those which are difficult to observe. This change of coordinates amounts to an equivalence transformation of the realization (A, B, C, D) of (1) called state-space transformation as in (7), where T now is the matrix representing the change of coordinates. The new system matrices $(T^{-1}AT, T^{-1}B, CT, D)$ form a balanced realization of (1). Truncating in this balanced realization the states that are hard to reach and difficult to observe results in a ROM.

Consider the Lyapunov equations

$$AP + PA^T + BB^T = 0, \ \ A^T Q + QA + C^T C = 0. \tag{9}$$

The solution matrices P and Q are called controllability and observability Gramians, respectively. If both Gramians are positive definite, the LTI system is minimal. This will be assumed from here on in this section.

In balanced coordinates the Gramians P and Q of a stable minimal LTI system satisfy $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with the Hankel singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$. The Hankel singular values are the positive square roots of the eigenvalues of the product of the Gramians PQ, $\sigma_k = \sqrt{\lambda_k(PQ)}$. They are system invariants, i.e., they are independent of the chosen realization of (1) as they are preserved under state-space transformations.

Given the LTI system (1) in a non-balanced coordinate form and the Gramians P and Q satisfying (9), the transformation matrix T which yields an LTI system in balanced coordinates can be computed via the so-called square root algorithm as follows:

- Compute the Cholesky factors S and R of the Gramians such that $P = S^T S$, $Q = R^T R$.
- Compute the singular value decomposition of $SR^T = \Phi \Sigma \Gamma^T$, where Φ and Γ are orthogonal matrices and Σ is a diagonal matrix with the Hankel singular values on its diagonal. $T = S^T \Phi \Sigma^{-\frac{1}{2}}$ yields the balancing transformation (note that $T^{-1} = \Sigma^{\frac{1}{2}} \Phi^T S^{-T} = \Sigma^{-\frac{1}{2}} \Gamma^T R$).
- Partition Φ, Σ, Γ into blocks of corresponding sizes,

$$\Sigma = \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}, \Phi = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \Gamma^T = \begin{bmatrix} \Gamma_1^T \\ \Gamma_2^T \end{bmatrix},$$

with $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ and apply T to (1) to obtain (7) with

$$T^{-1}AT = \begin{bmatrix} W^T A V & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \ T^{-1}B = \begin{bmatrix} W^T B \\ B_2 \end{bmatrix}, \tag{10}$$

and $CT = [CV \ C_2]$ for $W = R^T \Gamma_1 \Sigma_1^{-\frac{1}{2}}$ and $V = S^T \Phi_1 \Sigma_1^{-\frac{1}{2}}$. Preserving the r dominant Hankel singular values by truncating the rest yields the reduced-order model as in (5).

As $W^T V = I$, balanced truncation is an oblique projection method. The reduced-order model is stable with the Hankel singular values $\sigma_1, \ldots, \sigma_r$. It can be shown that if $\sigma_r > \sigma_{r+1}$, the error bound

$$\|G(\cdot) - \widetilde{G}(\cdot)\|_{\mathcal{H}_\infty} \leq 2 \sum_{k=r+1}^{n} \sigma_k \tag{11}$$

holds. Given an error tolerance, this allows to choose the appropriate order r of the reduced system in the course of the computations.

As the explicit computation of the balancing transformation T is numerically hazardous, one usually uses the equivalent balancing-free square root algorithm (Varga 1991) in which orthogonal bases for the column spaces of V and W are computed. The so obtained ROM is no longer balanced, but preserves all other properties (error bound, stability). Furthermore, it is shown in Benner et al. (2000) how to implement the balancing-free square root algorithm using low-rank approximations to S and R without ever

M

having to resort to the square solution matrices P and Q of the Lyapunov equations (9). This yields an efficient algorithm for balanced truncation for LTI systems with large dense matrices. For systems with large-scale sparse A efficient algorithms based on sparse solvers for (9) exist; see Benner (2006).

By replacing the solution matrices P and Q of (9) by other pairs of positive (semi-)definite matrices characterizing alternative controllability and observability related system information, one obtains a family of model reduction methods including stochastic/bounded-real/positive-real balanced truncation. These can be used if further properties like minimum phase, passivity, etc. are to be preserved in the reduced-order model; for further details, see Antoulas (2005) and Obinata and Anderson (2001).

The balanced truncation yields good approximation at high frequencies as $\widetilde{G}(\iota\omega) \to G(\iota\omega)$ for $\omega \to \infty$ (as $\widetilde{D} = D$), while the maximum error is often attained for $\omega = 0$. For a perfect match at zero and a good approximation for low frequencies, one may employ the *singular perturbation approximation* (SPA, also called balanced residualization). In view of (7) and (10), balanced truncation can be seen as partitioning $T^{-1}x$ according to (10) into $[x_1^T, x_2^T]^T$ and setting $x_2 \equiv 0$ (i.e., $\dot{x}_2 = 0$ as well). For SPA, one only sets $\dot{x}_2 = 0$, such that

$$\dot{x}_1 = WTAVx_1 + A_{12}x_2 + WTBu,$$

$$0 = A_{21}x_1 + A_{22}x_2 + B_2u.$$

Solving the second equation for x_2 and inserting it into the first equation yields

$$\dot{x}_1 = \left(WTAV - A_{12}A_{22}^{-1}A_{21}\right)x_1$$
$$+ \left(WTB - A_{12}A_{22}^{-1}B_2\right)u.$$

For the output equation, it follows

$$\tilde{y} = \left(CV - C_2A_{22}^{-1}A_{21}\right)x_1 + \left(D - C_2A_{22}^{-1}B_2\right)u.$$

This reduced-order model makes use of the information in the matrices A_{12}, A_{21}, A_{22}, B_2, and C_2 discarded by balanced truncation. It fulfills

$\widetilde{G}(0) = G(0)$ and the error bound (11); moreover, it preserves stability.

Besides SPA, another related truncation method that is not based on projection is *optimal Hankel norm approximation* (HNA). The description of HNA is technically quite involved; for details, see Zhou et al. (1996) and Glover (1984). It should be noted that the so obtained ROM usually has similar stability and accuracy properties as for balanced truncation.

Interpolation-Based Methods

Another family of methods for MOR is based on (rational) interpolation. The unifying feature of the methods in this family is that the original TFM (2) is approximated by a rational matrix function of lower degree satisfying some interpolation conditions (i.e., the original and the reduced-order TFM coincide, e.g., $G(s_0) = \widetilde{G}(s_0)$ at some predefined value s_0 for which $A - s_0E$ is nonsingular). Computationally this is usually realized by certain Krylov subspace methods.

The classical approach is known under the name of *moment-matching* or *Padé(-type) approximation*. In these methods, the transfer functions of the original and the reduced order systems are expanded into power series, and the reduced-order system is then determined so that the first coefficients in the series expansions match. In this context, the coefficients of the power series are called moments, which explains the term moment matching.

Classically the expansion of the TFM (2) in a power series about an expansion point s_0

$$G(s) = \sum_{j=0}^{\infty} M_j(s_0)(s - s_0)^j \qquad (12)$$

is used. The moments $M_j(s_0)$, $j = 0, 1, 2, \ldots$, are given by

$$M_j(s_0) = -C\left[(A - s_0E)^{-1}E\right]^j(A - s_0E)^{-1}B.$$

Consider the (block) Krylov subspace $\mathcal{K}_k(F, H) = \text{span}\{H, FH, F^2H, \ldots, F^{k-1}H\}$ for $F = (A - s_0E)^{-1}E$ and $H = -(A - s_0E)^{-1}B$ with

an appropriately chosen expansion point s_0 which may be real or complex. From the definitions of A, B, and E, it follows that $F \in \mathbb{K}^{n \times n}$ and $H \in \mathbb{K}^{n \times m}$, where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$ depending on whether s_0 is chosen in $\mathbb{R}$ or in $\mathbb{C}$. Considering $\mathcal{K}_k(F, H)$ column by column, this leads to the observation that the number of column vectors in $\{H, FH, F^2 H, \ldots, F^{k-1} H\}$ is given by $r = m \cdot k$, as there are k blocks $F^j H \in \mathbb{K}^{n \times m}$, $j = 0, \ldots, k - 1$. In the case when all r column vectors are linearly independent, the dimension of the Krylov subspace $\mathcal{K}_k(F, H)$ is $m \cdot k$. Assume that a unitary basis for this block Krylov subspace is generated such that the column space of the resulting unitary matrix $V \in \mathbb{C}^{n \times r}$ spans $\mathcal{K}_k(F, G)$. Applying the Galerkin projection $\Pi = V V^*$ to (1) yields a reduced system whose TFM satisfies the following (Hermite) interpolation conditions at s_0:

$$\widetilde{G}^{(j)}(s_0) = G^{(j)}(s_0), \quad j = 0, 1, \ldots, k - 1.$$

That is, the first $k - 1$ derivatives of G and $\widetilde{G}$ coincide at s_0. Considering the power series expansion (12) of the original and the reduced-order TFM, this is equivalent to saying that at least the first k moments $\widetilde{M}_j(s_0)$ of the transfer function $\widetilde{G}(s)$ of the reduced system (3) are equal to the first k moments $M_j(s_0)$ of the TFM $G(s)$ of the original system (1) at the expansion point s_0:

$$M_j(s_0) = \widetilde{M}_j(s_0), \quad j = 0, 1, \ldots, k - 1.$$

If further the r columns of the unitary matrix W span the block Krylov subspace $\mathcal{K}_k(F, H)$ for $F = (A - s_0 E)^{-T} E$ and $H = -(A - s_0 E)^{-T} C^T$, applying the Petrov-Galerkin projection $\Pi = V(W^* V)^{-1} W^*$ to (1) yields a reduced system whose TFM matches at least the first $2k$ moments of the TFM of the original system.

Theoretically, the matrix V (and W) can be computed by explicitly forming the columns which span the corresponding Krylov subspace $\mathcal{K}_k(F, H)$ and using the Gram-Schmidt algorithm to generate unitary basis vectors for $\mathcal{K}_k(F, H)$. The forming of the moments (the Krylov subspace blocks $F^j H$) is numerically precarious and has to be avoided under all

circumstances. Using Krylov subspace methods to achieve an interpolation-based ROM as described above is recommended. The unitary basis of a (block) Krylov subspace can be computed by employing a (block) Arnoldi or (block) Lanczos method; see, e.g., Antoulas (2005), Golub and Van Loan (2013), and Freund (2003).

In the case when an oblique projection is to be used, it is not necessary to compute two unitary bases as above. An alternative is then to use the nonsymmetric Lanczos process (Golub and Van Loan 2013). It computes bi-unitary (i.e., $W^* V = I_r$) bases for the above mentioned Krylov subspaces and the reduced-order model as a by-product of the Lanczos process. An overview of the computational techniques for moment-matching and Padé approximation summarizing the work of a decade is given in Freund (2003) and the references therein.

In general, the discussed MOR approaches are instances of rational interpolation. When the expansion point is chosen to be $s_0 = \infty$, the moments are called Markov parameters and the approximation problem is known as partial realization. If $s_0 = 0$, the approximation problem is known as Padé approximation.

As the use of one single expansion point s_0 leads to good approximation only close to s_0, it might be desirable to use more than one expansion point. This leads to multipoint moment-matching methods, also called rational Krylov methods; see, e.g., Ruhe and Skoogh (1998), Antoulas (2005), and Freund (2003).

In contrast to balanced truncation, these (rational) interpolation methods do not necessarily preserve stability. Remedies have been suggested; see, e.g., Freund (2003).

The use of complex-valued expansion points will lead to a complex-valued reduced-order system (3). In some applications (in particular, in case the original system is real valued), this is undesired. In that case one can always use complex-conjugate pairs of expansion points as then the entire computations can be done in real arithmetic.

The methods just described provide good approximation quality locally around the expansion

points. They do not aim at a global approximation as measured by the $\mathcal{H}_2$- or $\mathcal{H}_\infty$-norms. In Gugercin et al. (2008), an iterative procedure is presented which determines locally optimal expansion points w.r.t. the $\mathcal{H}_2$-norm approximation under the assumption that the order r of the reduced model is prescribed and only 0th- and 1st-order derivatives are matched. Also, for multi-input multi-output systems (i.e., m and p in (1) are both larger than one), no full moment matching is achieved, but only tangential interpolation: $G(s_j)b_j = \widetilde{G}(s_j)b_j$, $c_j^* G(s_j) = c_j^* \widetilde{G}(s_j)$, $c_j^* G'(s_j)b_j = c_j^* \widetilde{G}'(s_j)b_j$, for certain vectors b_j, c_j determined together with the optimal s_j by the iterative procedure.

Tools

Almost all commercial software packages for structural dynamics include modal analysis/truncation as a means to compute a ROM. Modal truncation and balanced truncation are available in the MATLAB® Control System Toolbox and the MATLAB® Robust Control Toolbox.

Numerically reliable, well-tested, and efficient implementations of many variants of balancing-based MOR methods as well as Hankel norm approximation and singular perturbation approximation can be found in the Subroutine Library In Control Theory (SLICOT, http://www.slicot.org) (Varga 2001). Easy-to-use MATLAB interfaces to the Fortran 77 subroutines from SLICOT are available in the SLICOT Model and Controller Reduction Toolbox (http://slicot.org/matlab-toolboxes/basic-control); see Benner et al. (2010). An implementation of moment matching via the (block) Arnoldi method is available in MOR for ANSYS®(http://modelreduction.com/Software.html).

There exist benchmark collections with mainly a number of LTI systems from various applications. There one can find systems in computer-readable format which can easily be used to test new algorithms and software:

- Oberwolfach Model Reduction Benchmark Collection
  http://simulation.uni-freiburg.de/downloads/benchmark/
- NICONET Benchmark Examples
  http://www.icm.tu-bs.de/NICONET/benchmodred.html

The MOR WiKi http://morwiki.mpi-magdeburg.mpg.de/morwiki/ is a platform for MOR research and provides discussions of a number of methods, links to further software packages (e.g., MOREMBS and MORPACK), as well as additional benchmark examples.

Summary and Future Directions

MOR of LTI systems can now be considered as an established computational technique. Some open issues still remain and are currently investigated. These include methods yielding good approximation in finite frequency or time intervals. Though numerous approaches for these tasks exist, methods with sharp local error bounds are still desirable. A related problem is the reduction of closed-loop systems and controller reduction. Also, the generalization of the methods discussed in this essay to descriptor systems (i.e., systems with DAE dynamics), second-order systems, or unstable LTI systems has only been partially achieved. An important problem class getting a lot of current attention consists of (uncertain) parametric systems. Here it is important to preserve parameters as symbolic quantities in the ROM. Most of the current approaches are based in one way or another on interpolation. MOR for nonlinear systems has also been a research topic for decades. Still, the development of satisfactory methods in the context of control design having computable error bounds and preserving interesting system properties remains a challenging task.

Cross-References

▶ Basic Numerical Methods and Software for Computer Aided Control Systems Design
▶ Multi-domain Modeling and Simulation

Bibliography

Antoulas A (2005) Approximation of large-scale dynamical systems. SIAM, Philadelphia

Benner P (2006) Numerical linear algebra for model reduction in control and simulation. GAMM Mitt 29(2):275–296

Benner P, Quintana-Ortí E, Quintana-Ortí G (2000) Balanced truncation model reduction of large-scale dense systems on parallel computers. Math Comput Model Dyn Syst 6:383–405

Benner P, Mehrmann V, Sorensen D (2005) Dimension reduction of large-scale systems. Lecture Notes in Computational Science and Engineering, vol 45. Springer, Berlin/Heidelberg

Benner P, Kressner D, Sima V, Varga A (2010) Die SLICOT-Toolboxen für Matlab (The SLICOT-Toolboxes for Matlab) [German]. at-Automatisierungstechnik 58(1):15–25. English version available as SLICOT working note 2009-1, 2009, http://slicot.org/working-notes/

Benner P, Hochstenbach M, Kürschner P (2011) Model order reduction of large-scale dynamical systems with Jacobi-Davidson style eigensolvers. In: Proceedings of the International Conference on Communications, Computing and Control Applications (CCCA), March 3-5, 2011 at Hammamet, Tunisia, IEEE Publications (6 pages)

Freund R (2003) Model reduction methods based on Krylov subspaces. Acta Numer 12:267–319

Glover K (1984) All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. Internat J Control 39:1115–1193

Golub G, Van Loan C (2013) Matrix computations, 4th edn. Johns Hopkins University Press, Baltimore

Gugercin S, Antoulas AC, Beattie C (2008) $\mathcal{H}_2$ model reduction for large-scale dynamical systems. SIAM J Matrix Anal Appl 30(2):609–638

Obinata G, Anderson B (2001) Model reduction for control system design. Communications and Control Engineering Series. Springer, London

Ruhe A, Skoogh D (1998) Rational Krylov algorithms for eigenvalue computation and model reduction. Applied Parallel Computing. Large Scale Scientific and Industrial Problems, Lecture Notes in Computer Science, vol 1541. Springer, Berlin/Heidelberg, pp 491–502

Schilders W, van der Vorst H, Rommes J (2008) Model order reduction: theory, research aspects and applications. Springer, Berlin/Heidelberg

Varga A (1991) Balancing-free square-root algorithm for computing singular perturbation approximations. In: Proceedings of the 30th IEEE CDC, Brighton, pp 1062–1065

Varga A (1995) Enhanced modal approach for model reduction. Math Model Syst 1(2):91–105

Varga A (2001) Model reduction software in the SLICOT library. In: Datta B (ed) Applied and computational control, signals, and circuits. The Kluwer International Series in Engineering and Computer Science, vol 629. Kluwer Academic, Boston, pp 239–282

Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River, NJ

Model Reference Adaptive Control

Jing Sun
University of Michigan, Ann Arbor, MI, USA

Synonyms

MRAC

Abstract

The fundamentals and design principles of model reference adaptive control (MRAC) are described. The controller structure and adaptive algorithms are delineated. Stability and convergence properties are summarized.

Keywords

Certainty equivalence; Lyapunov-SPR design; MIT rule

Introduction

Model reference adaptive control (MRAC) is an important adaptive control approach, supported by rigorous mathematical analysis and effective design toolsets. It is made up of a feedback control law that contains a controller $C(s, \theta_c)$ and an adjustment mechanism that generates the controller parameter updates $\theta_c(t)$ online. While different MRAC configurations can be found in the literature, the structure shown in Fig. 1 is commonly used and includes all the basic components of an MRAC system. The prominent features of MRAC are that it incorporates a reference model which represents the desired

M
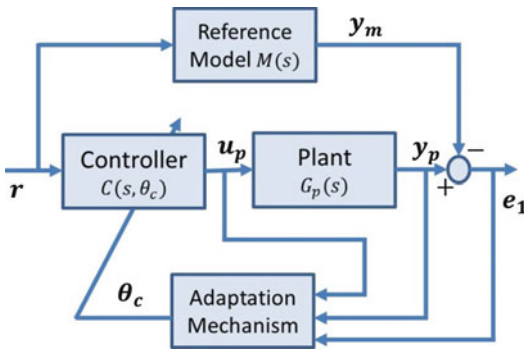
Model Reference Adaptive Control, Fig. 1 Schematic of MRAC

input–output behavior and that the controller and adaptation law are designed to force the response of the plant, y_p, to track that of the reference model, y_m, for any given reference input r.

Different approaches have been used to design MRAC, and each may lead to a different implementation scheme. The implementation schemes fall into two categories: direct and indirect MRAC. The former updates the controller parameters θ_c directly using an adaptive law, while the latter updates the plant parameters θ_p first using an estimation algorithm and then updates θ_c by solving, at each time t, certain algebraic equations that relate θ_c with the online estimates of θ_p. In both direct and indirect MRAC schemes, the controller structure is kept the same as that which would be used in the case that the plant parameters are known.

MRC Controller Structure

Consider the design objective of model reference control (MRC) for linear time-invariant systems: Given a reference model $M(s)$, find a control law such that the closed-loop system is stable and $y_p \to y_m$ as $t \to \infty$ for any bounded reference signal r.

For the case of known plant parameters, the MRC objective can be achieved by designing the controller so that the closed-loop system has a transfer function equal to $M(s)$. This

is the so-called model matching condition. To assure the existence of a causal controller that meets the model matching condition and guarantees internal stability of the closed-loop system, the following assumptions are essential:

- A1. The plant has a stable inverse, and the reference model is chosen to be stable.
- A2. The relative degree of $M(s)$ is equal to or greater than that of the plant $G_p(s)$. Herein, the relative degree of a transfer function refers to the difference between the orders of the denominator and numerator polynomials.

It should be noted that these assumptions are imposed to the MRC problem so that there is enough structural flexibility in the plant and in the reference model to meet the control objectives. A1 is necessary for maintaining internal stability of the system while meeting the model matching condition, and A2 is needed to ensure the causality of the controller. Both assumptions are essential for non-adaptive applications when the plant parameters are known, let alone for the adaptive cases when the plant parameters are unknown.

The reference model plays an important role in MRAC, as it will define the feasibility of MRAC design as well as the performance of the resulting closed-loop MRAC system. The reference model should reflect the desired closed-loop performance. Namely, any time-domain or frequency-domain specifications, such as time constant, damping ratio, natural frequency, bandwidth, etc., should be properly reflected in the chosen transfer function $M(s)$.

The controller structure for the MRAC is derived with these assumptions for the known plant case and extended to the adaptive case by combining it with a proper adaptive law. Under assumptions A1–A2, there exist infinitely many control solutions C that will achieve the MRC design objective for a given plant transfer function $G_p(s)$. Nonetheless, only those extendable to MRAC with the simplest structure are of interest. It is known that if a special controller structure with the following parametrization is imposed, then the solution to the model matching

condition, in terms of the ideal parameter θ_c^*, will be unique:

$$u_p = \theta_1^{*T}\omega_1 + \theta_2^{*T}\omega_2 + \theta_3^{*T}y_p + c_0^* r = \theta_c^{*T}\omega$$

where $\theta_c^* \in R^{2n}$, n is the order of the plant,

$$\theta_c^* = \begin{bmatrix} \theta_1^* \\ \theta_2^* \\ \theta_3^* \\ c_0^* \end{bmatrix}, \omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ y_p \\ r \end{bmatrix}$$

and $\omega_1, \omega_2 \in R^{n-1}$ are signals internal to the controller generated by stable filters (Ioannou and Sun 1996).

This MRC control structure is particularly appealing for adaptive control development, as the parameters appear linearly in the control law expression, leading to a convenient linear parametric model for adaptive algorithm development.

Adaptation Algorithm

Design of adaptive algorithms for parameter updating can be pursued in several different approaches, thereby resulting in different MRAC schemes. Three direct design approaches, namely, the Lyapunov-SPR, the certainty equivalence, and the MIT rule, will be briefly described together with indirect MRAC.

Lyapunov-SPR Design

One popular MRAC algorithm is derived using Lyapunov's direct method and the Meyer-Kalman-Yakubovich (MKY) Lemma based on the strictly positive real (SPR) argument. The concept of SPR transfer functions originates from network theory and is related to the driving point impendence of dissipative networks. The MKY Lemma states that given a stable transfer function $M(s)$ and its realization (A, B, C, d) where $d \geq 0$ and all eigenvalues of the matrix A are in the open left half plane: If $M(s)$ is SPR, then for any given positive definite matrix $L = L^T > 0$, there exists a scalar $\nu > 0$, a vector q, and a $P = P^T > 0$ such that

$$A^T P + PA = -qq^T - \nu L$$
$$PB - C = \pm q\sqrt{2d}$$

By choosing $M(s)$ to be SPR, one can formulate a Lyapunov function consisting of the state tracking and parameter estimation errors and use the MKY Lemma to define the adaptive law that will force the derivative of the Lyapunov function to be semi-negative definite. The resulting adaptive law has the following simple form:

$$\dot{\theta} = -\Gamma e_1 \omega \text{sign}(c_0^*)$$

where $e_1 = y_p - y_m$ is simply the tracking error and $c_0^* = k_m/k_p$ with k_m, k_p being the high frequency gain of the transfer function for the reference model $M(s)$ and the plant $G_p(s)$, respectively. This algorithm, however, applies only to systems with relative degree equal to 0 or 1, which is implied by the SPR condition imposed on $M(s)$ and assumption A2.

The Lyapunov-SPR-based MRAC design is mathematically elegant in its stability analysis but is restricted to a special class of systems. While it can be extended to more general cases with relative degrees equal to 2 and 3, the resulting control law and adaptive algorithm become much more complicated and cumbersome as efforts must be made to augment the control signal in such a way that the MKY Lemma is applicable to the "reformulated" reference model.

Certainty Equivalence Design

For more general cases with a high relative degree, another design approach based on "certainty equivalence" (CE) principle is preferred, due to the simplicity in its design as well as its robustness properties in the presence of modeling errors. This approach treats the design of the adaptive law as a parameter estimation problem, with the estimated parameters being the controller parameter vector θ_c^*. Using the specific linear formulation of the control law and assuming that θ_c^* satisfies the model matching condition, one can show that the ideal controller parameter satisfies the following parametric equation:

$$z = \theta_c^{*T} \omega_p$$

with

$$z = M(s)u_p, \omega_p = \begin{bmatrix} M(s)\omega_1 \\ M(s)\omega_2 \\ M(s)y_p \\ y_p \end{bmatrix}$$

This parametric model allows one to derive adaptive laws to estimate the unknown controller parameter θ_c^* using standard parameter identification techniques, such as the gradient and least squares algorithms. The corresponding MRAC is then implemented in the CE sense where the unknown parameters are replaced by their estimated value. It should be noted that a CE design does not guarantee closed-loop stability of the resulting adaptive system, and additional analysis has been carried out to establish closed-loop stability.

MIT Rule

Besides the Lyapunov-SPR and CE approaches mentioned earlier, the direct MRAC problem can also be approached using the so-called MIT rule, an early form of MRAC developed in the 1950s–1960s in the Instrumentation Laboratory at MIT for flight control. The designer defines a cost function, e.g., a quadratic function of tracking error, and then adjusts parameters in the direction of steepest descent. The negative gradient of the cost function is usually calculated through the sensitivity derivative approach. The formulation is quite flexible, as different forms of MIT rule can be derived by changing the cost function following the same procedure and reusing the same sensitivity functions. Despite its effectiveness in some practical applications, MRAC systems designed with MIT rule have had stability and robustness issues.

Indirect MRAC

While most of the MRAC systems are designed as direct adaptive systems, indirect MRAC systems can also be developed which explicitly estimate the plant parameter θ_p as an intermediate step. The adaptive law for an indirect MRAC includes two basic components: one for estimating the plant parameters and another for calculating the controller parameters based on the estimated plant parameters. This approach would be preferred if the plant transfer function is partially known, in which case the identification of the remaining unknown parameters represents a less complex problem. For example, if the plant has no zeros, the indirect scheme estimates $n + 1$ parameters, while the direct scheme has to estimate $2n$ parameters.

Indirect MRAC is a CE-based design. As such, the design is intuitive but the design process does not guarantee closed-loop stability, and separate analysis has to be carried out to establish stability. Except for systems with a low number of zeros, the "feasibility" problem could also complicate the matter, in the sense that the MRC problem may not have a solution for the estimated plant at some time instants even though the solution exists for the real plant. This problem is unique to the indirect design, and several mitigating solutions have been found at the expense of more complicated adaptation or control algorithms.

Stability, Robustness, and Parameter Convergence

Stability for MRAC often refers to the properties that all signals are bounded and tracking error converges to zero asymptotically. Robustness for adaptive systems implies that signal boundedness and tracking error convergence (to a small residue set) will be preserved in the presence of small perturbations such as disturbances, un-modeled dynamics, and time-varying parameters. For different MRAC schemes, different approaches are used to establish their properties.

For the Lyapunov-SPR-based MRAC systems, stability is established in the design process where the adaptive law is derived to enforce a Lyapunov stability condition. For CE-based designs, establishing stability for the closed-loop MRAC system is a nontrivial exercise for both direct and indirect schemes. Using properly normalized adaptive laws for parameter

estimation, however, stability can be proved for direct and indirect MRAC schemes. For MRAC systems designed with the MIT rule, local stability can be established under more restrictive conditions, such as when the parameters are close to the ideal ones.

It should be noted that the adaptive control algorithm in the original form has been shown to have robustness issues, and extensive publications in the 1980s and 1990s were devoted to robust adaptive control in attempts to mitigate the problem. Many modifications have been proposed and shown to be effective in "robustifying" the MRAC; interested readers are referred to the article on ▶ Robust Adaptive Control for more details.

Parameter convergence is not an intrinsic requirement for MRAC, as tracking error convergence can be achieved without parameter convergence. It has been shown, however, that parameter convergence could enhance robustness, particularly for indirect schemes. As in the case for parameter identification, a persistent excitation (PE) condition needs to be imposed on the regression signal to assure parameter convergence in MRAC. In general, PE is accomplished by properly choosing the reference input r. It can be established for most MRAC approaches that parameter convergence is achieved if, in addition to conditions required for stability, the reference input r is sufficiently rich of order $2n$, $\dot{r}$ is bounded, and there is no pole-zero cancelation in the plant transfer function. A signal is called to be sufficiently rich of order m if it contains at least $m/2$ distinct frequencies.

Summary and Future Directions

MRAC incorporates a reference model to capture the desired closed-loop responses and designs the control law and adaptation algorithm to force the output of the plant to follow the output of the reference model. Several different design approaches are available. Stability, robustness, and parameter convergence have been established for different MRAC designs with appropriate assumptions.

MRAC had been a very active and fruitful research topic from the 1960s to 1990s, and it formed important foundations for modern adaptive control theory. It also found many successful applications ranging from chemical process controls to automobile engine controls. More recent efforts have been mostly devoted to integrating it with other design approaches to treat nonstandard MRAC problems for nonlinear and complex dynamic systems.

Cross-References

▶ Adaptive Control of Linear Time-Invariant Systems
▶ Adaptive Control, Overview
▶ History of Adaptive Control
▶ Robust Adaptive Control

Recommended Reading

MRAC has been well covered in several textbooks and research monographs. Astrom and Wittenmark (1994) presented different MRAC schemes in a tutorial fashion. Narendra and Annaswamy (1989) focused on stability of deterministic MRAC systems. Ioannou and Sun (1996) covered the detailed derivation and analysis of different MRAC schemes and provided a unified treatment for their stability and robustness analysis. MRAC systems for discrete-time (Goodwin and Sin 1984) and for nonlinear (Krstic et al. 1995) processes are also well explored.

Bibliography

Astrom KJ, Wittenmark B (1994) Adaptive control. Second edition. Prentice Hall, Englewood Cliffs

Goodwin GC, Sin KS (1984) Adaptive filtering, prediction and control. Prentice Hall, Englewood Cliffs

Ioannou PA, Sun J (1996) Robust adaptive control. Prentice Hall, Upper Saddle River

Krstic K, Kanellakopoulos I, Kokotovic PV (1995) Nonlinear and adaptive control design. Wiley, New York

Narendra KS, Annaswamy AM (1989) Stable adaptive systems. Prentice Hall, Englewood Cliffs

M

Model-Based Performance Optimizing Control

Sebastian Engell
Fakultät Bio- und Chemieingenieurwesen,
Technische Universität Dortmund, Dortmund,
Germany

Abstract

In many applications, e.g., in chemical process control, the purpose of control is to achieve an optimal performance of the controlled system despite the presence of significant uncertainties about its behavior and of external disturbances. Tracking of set points is often required for lower-level control loops, but at the system level in most cases, this is not the primary concern and may even be counterproductive. In this entry, the use of dynamic online optimization on a moving finite horizon to realize optimal system performance is discussed. By real-time optimization, a performance-oriented or economic cost criterion is minimized or maximized over a finite horizon while the usual control specifications enter as constraints but not as set points. This approach integrates the computation of optimal set-point trajectories and of the regulation to these trajectories.

Keywords

Model-predictive control (MPC); Integrated optimization and control; Real-time optimization (RTO); Performance optimizing control; Process control

Introduction

From a systems point of view, the purpose of automatic feedback control (and that of manual control as well) in many cases is *not* primarily to keep the controlled variables at their set points as well as possible or to track dynamic set-point changes but to operate the system such that its *performance* is optimized in the presence of disturbances und uncertainties, exploiting the information gained in real time from the available measurements. This holds generally for the higher control layers in the process industries but similarly for many other applications. Suppose that, for example, the availability of cooling water at a lower temperature than assumed as a worst case during plant design enables plant operation at a higher throughput. In this case, what sense does it make to enforce the nominal operating point by tight feedback control? For a combustion engine, the goal is to achieve the desired torque with minimum consumption of fuel. For a cooling system, the goal is to keep the temperature of the goods or of a room within certain bounds with minimum consumption of energy, possibly weighted against the wear of the equipment. To regulate some variables to their set points may help to achieve these goals but it is not the real performance target for the overall system. Feedback control loops therefore usually are part of control hierarchies that establish good performance of the overall system and the meeting of constraints on its operation.

There are four main approaches to the integration of feedback control with system performance optimization:

- Choice of regulated variables such that, implicitly via the regulation of these variables to their set points, the performance of the overall system is close to optimal (see the chapter on ▸ Control Structure Selection).
- Tracking of necessary conditions of optimality where variables which determine the optimal operating policy are kept at or close to their constraints. This is a widespread approach especially in chemical batch processes where, e.g., the feeding of reactants is such that the maximum cooling power available is used (Finkler et al. 2014); see also the chapter on ▸ Control and Optimization of Batch Processes).

In these two approaches, the choice of the optimal set points or constraints to be tracked is done off-line, and they are then implemented by the feedback layer of the process control hierarchy (see the chapter on ▸ Control Hierarchy of Large Processing Plants: An Overview).

- Combination of a regulatory (tracking) feedback control with an optimization of the set points or system trajectories (called real-time optimization in the process industries) (see the chapter on ▶ Real-Time Optimization of Industrial Processes).
- Reformulation of model-predictive control such that the control target is not the tracking of references but the optimization of the system performance over a finite horizon, taking constraints of system variables or inputs into account directly within the online optimization. Here, the optimization is performed with a dynamic model, in contrast to the steady-state optimization in real-time optimization or in the choice of self-optimizing control structures.

The first three approaches are currently state of the art in the process industries. Tracking of necessary conditions of optimality is usually designed based on process insight rather than based upon a rigorous analysis, and the same holds for the selection of regulatory control structures. The last one is the most challenging approach in terms of the required models and algorithms and computing power, and its theoretical foundations are still under development. But on the other hand, it also has the highest potential in terms of the resulting performance of the controlled system, and it is structurally simple and easier to tune because the natural performance specification does not have to be translated into controller tunings, weights, etc. Therefore, the idea of direct model-based performance optimizing control has found much attention in process control in recent years.

The four approaches above are discussed in more detail below. We also provide some historical notes and outline some areas of continuing research.

Performance Optimization by Regulation to Fixed Set Points

Morari et al. (1980) stated that the objective in the synthesis of a control structure is "to translate the economic objectives into process control objectives." A subgoal in this "translation" is to select the regulatory control structure of a process such that steady-state optimality of process operations is realized to the maximum extent possible by driving the selected controlled variables to suitably chosen set points. A control structure with this property was termed "self-optimizing control" by Skogestad (2000). It should adjust the manipulated variables by keeping a function of the measured variables constant such that the process is operated at the economically optimal steady state in the presence of disturbances. From a system point of view, a control structure that yields nice transient responses and tight control of the selected variables may be of little use or even counterproductive if keeping the regulated variables at their set points does not improve the performance of the system. Ideally, in the steady state, a similar performance is obtained as it would be realized by optimizing the stationary values of the operational degrees of freedom of the system for known disturbances d and a perfect model. By regulating the controlled variables to their set points at the steady state in the presence of disturbances, a mapping $u = f(y_{set}, d)$ is implicitly realized which should be an approximation of the performance optimizing inputs $u_{opt}(d)$. The choice of the self-optimizing control structure takes only the steady-state performance into account, not the dynamic reaction of the controlled plant. An extension of the approach to include also the dynamic behavior can be found in Pham and Engell (2011).

Tracking of Necessary Conditions of Optimality

Very often, the optimal operation of a system in a certain phase of its evolution or under certain conditions is defined by some variables being at their constraints. If these variables are known and the conditions can be monitored, a switching control structure can be built that keeps the (possibly changing) set of critical variables at their constraints despite inaccuracies of the model, external disturbances, etc. In fact it turns out that such control schemes can, in the case of varying

parameters and in the presence of disturbances, perform as good as sophisticated model-based optimization schemes (Finkler et al. 2013).

Performance Optimization by Steady-State Optimization and Regulation

A well-established approach to create a link between regulatory control and the optimization of the performance of a system is to compute the set points of the controllers by an optimization layer. In process operations, this layer is called real-time optimization (RTO) (see, e.g., Marlin and Hrymak (1997) and the references therein). An RTO system is a model-based, upper-level control system that is operated in closed loop and provides set points to the lower-level control systems in order to maintain the process operation as close as possible to the economic optimum. It usually comprises an estimation of the plant state and plant parameters from the measured data and an economic or otherwise performance-related optimization of the operating point using a detailed nonlinear steady-state model.

As the RTO system employs a stationary process model and the optimization is only performed if the plant is approximately in a steady state, the time between successive RTO steps must be large enough for the plant to reach a new steady state after the last commanded move. This structure is based upon a separation of concerns and of time-scales between the RTO system and the process control system. The RTO system optimizes the system economics on a medium timescale (shifts to days), while the control system provides tracking and disturbance rejection on shorter timescales from seconds to hours.

As an approximation to real-time optimization with a nonlinear rigorous plant model, in many MPC implementations nowadays, an optimization of the steady-state values based on the linear model that is used in the MPC controller is implemented. Then the gain matrix of the model must be estimated carefully to obtain good results.

Performance Optimizing Control

Model-predictive control has become the standard solution for demanding control problems in the process industries (Qin and Badgwell 2003) and increasingly is used also in other domains. The core idea is to employ a model to predict the effect of the future manipulated variables on the future controlled variables over a finite horizon and to use optimization to determine sequences of inputs which minimize a cost function over the so-called prediction horizon. In the unconstrained case with linear plant model and a quadratic cost function, the optimal control moves can be computed by a closed-form solution. When constraints on inputs, outputs, and possibly also state variables are present, for a quadratic cost function and linear plant model, the optimization problem becomes a quadratic program (QP) that has to be solved in real time.

When the system dynamics are nonlinear and linear models are only sufficiently accurate within narrow operation bands, as is the case in many chemical processes, nonlinear model predictive control which is based on nonlinear models of the process dynamics provides superior performance and therefore has met increasing interest both in theory and in practice. The classical formulation of nonlinear model-predictive tracking control (TC) is

$$\min_{u} \phi_{TC} \left(\bar{y}, u \right)$$

$$\phi_{TC} = \sum_{n=1}^{N} \left(\sum_{i=1}^{P} \gamma_{n,i} \left(y_{n,ref} \left(k - i \right) - \bar{y}_n \left(k + i \right) \right)^2 \right) + \sum_{l=1}^{R} \left(\sum_{j=1}^{M} \alpha_{l,j} \Delta u_l^2 (k + j) \right)$$

s.t.

$$x(i+1) = f(x(i), z(i), u(i), i), i = k, \ldots, k+P$$

$$0 = g(x(i), z(i), u(i), i), i = k, \ldots, k+P$$

$$y(i+1) = h(x(i+1), u(i)), i = k, \ldots, k+P$$

$$x_{min} \le x(i) \le x_{max}, i = k, \ldots, k+P$$

$$y_{min} \le \bar{y}(i) \le y_{max}, i = k, \ldots, k+P$$

$$u_{min} \le u(i) \le u_{max}, i = k, \ldots, k+M$$

$$-\Delta u_{min} \le \Delta u(i) \le \Delta u_{max}, i = k, \ldots, k+M$$

$$u(i) = u(i-1) + \Delta u(i), i = k, \ldots, k+M$$

$$u(i) = u(k+M), \forall i > k+M.$$

Here f and g represent the plant model in the form of a system of differential-algebraic Equations and h is the output function. P is the length of the prediction horizon and M is the length of the control horizon, and $y_1, \cdots, y_N$ are the predicted control outputs, $u_1, \cdots, u_R$ are the control inputs. α and γ represent the weights on the control inputs and the control outputs, respectively. y_{ref} refers to the set point or the desired output trajectory, and $\hat{y}(i)$ are the corrected model predictions. N is number of the controlled outputs, and R is the number of the control inputs. Compensation for plant-model mismatch and unmeasured disturbances is usually done using the bias correction equations:

$$d(k) = y^{meas}(k) - y(k),$$

$$\bar{y}(k+i) = y(k+i) + d(k), i = k, \ldots, k+P.$$

The idea of direct performance optimizing control (POC) is to replace this formulation by a performance-related objective function:

$$\min_u \phi_{POC}(y, u)$$
$$\phi_{POC} = \sum_{l=1}^{R} \left(\sum_{j=1}^{M} \alpha_{l,j} \Delta u_l^2(k+j) \right) - \left(\sum_{i=1}^{P} \beta_i \psi(k+i) \right).$$

Here $\psi(k+i)$ represents the value of the performance cost criterion at the time step $[k+i]$.

The optimization of the future control moves is subject to the same constraints as before. In addition, instead of reference tracking, constraints are formulated for all outputs that are critical for the operation of the system or its performance, e.g., product quality specifications or limitations of the equipment. In contrast to reference tracking, these constraints usually are one-sided (inequalities) or define operation bands. By this formulation, e.g., the production revenues can be maximized online over a finite horizon, considering constraints on product purities and waste stream impurities. Feedback enters into the computation by the initialization of the model with a new initial state that is estimated from the available measurements of system variables and by the bias correction. Thus, direct performance optimizing control realizes an online optimization of all operational degrees of freedom in a feedback structure without tracking of a priori fixed set points or reference trajectories. The regularization term that penalizes control moves is added to the purely economic objective function to obtain smoother solutions.

This approach has several advantages over a combined steady-state optimization/ linear MPC scheme:

- Immediate reaction to disturbances, no waiting for the plant to reach a steady state is required.
- "Overregulation" is avoided – no variables are forced to fixed set points and all degrees of freedom can be used to improve the (economic) performance of the plant.
- Performance goals and process constraints do not have to be mapped to a control cost that defines a compromise between different goals. In this way, the formulation of the optimization problem and the tuning are facilitated compared to achieving good performance by tuning of the weights of a tracking formulation.
- More constraints than available manipulated variables can be handled as well as more manipulated variables than variables that have to be regulated.
- No inconsistency arises from the use of different models on different layers.
- The overall scheme is structurally simple.

M

Similar to any NMPC controller that is designed for reference tracking, a successful implementation will require careful engineering such that as many uncertainties as possible are compensated by simple feedback controllers and only the key dynamic variables are handled by the optimizing controller based on a rigorous model of the essential dynamics and of the stationary relations of the plant without too much detail.

History and Examples

The idea of economic or performance optimizing control originated from the process control community. The first papers on directly integrating economic considerations into model-predictive control Zanin et al. (2000) proposed to achieve a better economic performance by adding an economic term to a classical tracking performance criterion and applied this to the control of a fluidized bed catalytic cracker. Helbig et al. (2000) discussed different ways to integrate optimization and feedback control including direct dynamic optimization for the example of a semi-batch reactor. Toumi and Engell (2004) and Erdem et al. (2004) demonstrated online performance optimizing control schemes for simulated moving bed (SMB) chromatographic separations in lab scale. SMB processes are periodic processes and constitute prototypical examples where additional degrees of freedom can be used to simultaneously optimize *system* performance and to meet product specifications. Bartusiak (2005) reported already industrial applications of carefully engineered performance optimizing NMPC controllers.

Direct performance optimizing control was suggested as a promising general new control paradigm for the process industries by Rolandi and Romagnoli (2005), Engell (2006, 2007), Rawlings and Amrit (2009), and others. Meanwhile it has been demonstrated in many simulation studies that direct optimization of a performance criterion can lead to superior economic performance compared to classical tracking (N)MPC, e.g., Ochoa et al. (2010) for a

bioethanol process and Idris and Engell (2012) for a reactive distillation column.

Further Issues

Modeling and Robustness

In a direct performance optimizing control approach, sufficiently accurate dynamic nonlinear process models are needed. While in the process industries, nonlinear steady-state models are nowadays available for many processes because they are built and used extensively in the process design phase, there is still a considerable additional effort required to formulate, implement, and validate nonlinear dynamic process models. The effort for rigorous or semi-rigorous modeling usually dominates the cost of an advanced control project. The alternative approach to use black-box or gray-box models as proposed frequently in nonlinear model-predictive control may be effective for regulatory control where the model only has to capture the essential dynamic features of the plant near an operating point, but it seems to be less suitable for optimizing control where the optimal plant performance is aimed at and hence the best stationary values of the inputs and of the controlled variables have to be computed by the controller. As increasingly so-called operator training simulators are built in parallel to the construction of a plant and are continuously used and updated after the commissioning phase, it seems attractive to use the models contained in the simulators also for online optimization. However, the model formulations often are not suitable for this purpose.

Model inaccuracies always have to be taken into account. They not only lead to suboptimal performance but also can cause that the constraints even on measured variables cannot be met in the future because of an insufficient back-off from the constraints. A new approach to deal with uncertainties about model parameters and future influences on the process is multistage scenario-based optimization with recourse. Here the model uncertainties are represented by a set of scenarios of parameter variations and the future availability

of additional information is taken into account. It has been demonstrated that this is an effective tool to handle model uncertainties and to automatically generate the necessary back-off without being overly conservative (Lucia et al. 2013).

State Estimation

For the computation of economically optimal process trajectories based upon a rigorous nonlinear process model, the state variables of the system at the beginning of the prediction horizon must be known. As not all states will be measured in a practical application, state estimation is a key ingredient of a performance optimizing controller. Extended Kalman filters are the standard solution used in the process industries, if the nonlinearities are significant, unscented Kalman filters or particle filters may be used. A novel approach is to formulate the state estimation problem also as an optimization problem on a moving horizon (Rao et al. 2003). The estimation of some important varying unknown model parameters can be included in this formulation. As accurate state estimation is at least as critical for the performance of the closed-loop system as the exact tuning of the optimizing controller, more attention should be paid to the investigation of the performance of state estimation schemes in realistic situations with non-negligible model-plant mismatch.

Stability

Optimization of a cost function over a finite horizon in general neither assures optimality of the complete trajectory beyond this horizon nor stability of the closed-loop system. Closed-loop stability has been addressed extensively in the theoretical research in nonlinear model-predictive control. Stability can be assured by a proper choice of the stage cost within the prediction horizon and the addition of a cost on the terminal state and the restriction of the terminal state to a suitable set. In performance optimizing MPC, there is no a priori known steady state to which the trajectory should converge, and the economic cost function may not satisfy the usual conditions for closed-loop stability, e.g., because it only involves some of the inputs. In recent years, important results on closed-loop stability guaranteeing formulations have nonetheless been obtained, involving terminal constraints or a quasi-infinite horizon (Angeli et al. 2012; Diehl et al. 2011; Grüne 2013).

Reliability and Transparency

Nowadays quite large nonlinear dynamic optimization problems can be solved in real time, not only for slow processes as they are found in the chemical industry but also in mechatronics and automotive control. So this issue does no longer prohibit the application of a performance optimizing control scheme to complex systems. A practically very important limiting issue however is that of reliability and transparency. It is difficult to guarantee that a nonlinear optimizer will provide a solution which at least satisfies the constraints and gives a reasonable performance for all possible input data. While for an RTO scheme an inspection of the commanded set points by the operators usually will be feasible, this is less likely to be realistic in a dynamic situation. Hence, automatic result filters are necessary as well as a backup scheme that stabilizes the process in the case where the result of the optimization is not considered safe. In the process industries, the operators will continue to supervise the operation of the plant in the foreseeable future, so a control scheme that includes performance optimizing control must be structured into modules, the outputs of which can still be understood by the operators so that they build up trust in the optimization. Good operator interfaces that display the predicted moves and the predicted reaction of the plant and enable comparisons with the operators' intuitive strategies are believed to be essential for practical success.

Effort vs. Performance

The gain in performance by a more sophisticated control scheme always has to be traded against the increase in cost due to the complexity of the control scheme – a complex scheme will not only cause cost for its implementation, but it will need more maintenance by better qualified people than a simple one. If a carefully

chosen standard regulatory control layer leads to a close-to-optimal operation, there is no need for optimizing control. If the disturbances that affect profitability and cannot be handled well by the regulatory layer (in terms of economic performance) are slow, the combination of regulatory control and RTO is sufficient. In a more dynamic situation or for complex nonlinear multivariable plants, the idea of direct performance optimizing control should be explored and implemented if significant gains can be realized in simulations.

Cross-References

Bibliography

Angeli D, Amrit R, Rawlings J (2012) On average performance and stability of economic model predictive control. IEEE Trans Autom Control 57(7):1615–1626

Bartusiak RD (2005) NMPC: a platform for optimal control of feed- or product-flexible manufacturing. Preprints International workshop on assessment and future directions of NMPC, Freudenstadt, pp 3–14

Diehl M, Amrit R, Rawlings J, Angeli D (2011) A Lyapunov function for economic optimizing model predictive control. IEEE Trans Autom Control 56(3):703–707

Engell S (2006, 2007) Feedback control for optimal process operation. Plenary paper IFAC ADCHEM, Gramado, 2006; J Process Control 17:203–219

Erdem G, Abel S, Morari M, Mazzotti M, Morbidelli M (2004) Automatic control of simulated moving beds. Part II: nonlinear isotherms. Ind Eng Chem Res 43:3895–3907

Finkler T, Lucia S, Dogru M, Engell S (2013) A simple control scheme for batch time minimization of exothermic semi-batch polymerizations. Ind Eng Chem Res 52:5906–5920

Finkler TF, Kawohl M, Piechottka U, Engell S (2014) Realization of online optimizing control in an industrial semi-batch polymerization. J Process Control 24:399–414

Grüne L (2013) Economic receding horizon control without terminal constraints. Automatica 49:725–734

Helbig A, Abel O, Marquardt W (2000) Structural concepts for optimization based control of transient processes. In: Nonlinear model predictive control. Allgöwer F and Zheng A, eds. Birkhäuser, Basel pp 295–311

Idris IAN, Engell S (2012) Economics-based NMPC strategies for the operation and control of a continuous catalytic distillation process. J Process Control 22:1832–1843

Lucia S, Finkler T, Basak D, Engell S (2013) A new Robust NMPC Scheme and its application to a semi-batch reactor example. J Process Control 23:1306–1319

Marlin TE, Hrymak AN (1997) Real-time operations optimization of continuous processes. In: Proceedings of CPC V, Lake Tahoe, AIChE symposium series, vol 93, pp 156–164

Morari M, Stephanopoulos G, Arkun Y (1980) Studies in the synthesis of control structures for chemical processes, part I. AIChE J 26:220–232

Ochoa S, Wozny G, Repke J-U (2010) Plantwide optimizing control of a continuous bioethanol production process. J Process Control 20:983–998

Pham LC, Engell S (2011) A procedure for systematic control structure selection with application to reactive distillation. In: Proceedings of 18th IFAC world congress, Milan, pp 4898–4903

Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. Control Eng Pract 11:733–764

Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems. IEEE Trans Autom Control 48:246–258

Rawlings J, Amrit R (2009) Optimizing process economic performance using model predictive control In: Nonlinear model predictive control – towards new challenging applications. Springer, Berlin, pp 119–138

Rolandi PA, Romagnoli JA (2005) A framework for online full optimizing control of chemical processes. In: Proceedings of ESCAPE 15. Barcelona, Elsevier, pp 1315–1320

Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. J Process Control 10:487–507

Toumi A, Engell S (2004) Optimization-based control of a reactive simulated moving bed process for glucose isomerization. Chem Eng Sci 59:3777–3792

Zanin AC, Tvrzska de Gouvea M, Odloak D (2000) Industrial implementation of a real-time optimization strategy for maximizing production of LPG in a FCC unit. Comput Chem Eng 24:525–531

Modeling of Dynamic Systems from First Principles

S. Torkel Glad
Department of Electrical Engineering,
Linköping University, Linköping, Sweden

Abstract

This entry describes how models can be formed from the basic principles of physics and the other fields of science. Use can be made of similarities between different domains which leads to the concepts of bond graphs and, more abstractly, to port-controlled Hamiltonian systems. The class of models is naturally extended to differential algebraic equation (DAE) models. The concepts described here form a natural basis for parameter identification in gray box models.

Keywords

Introduction

The approach to the modeling of dynamic systems depends on how much is known about the system. When the internal mechanisms are known, it is natural to model them using known relationships from physics, chemistry, biology, etc. Often the result is a model of the following form:

$$\frac{dx}{dt} = f(x, u; \theta), \quad y = h(x, u; \theta) \quad (1)$$

where u is the input, y is the output, and the state x contains internal physical variables, while θ contains parameters. Typically all of these are vectors. The model is known as a state space model. In many cases some elements in θ are unknown and have to be determined using parameter estimation. When used in connection with system identification, these models are sometimes referred to as *gray box* models (in contrast to black box models) to indicate that some degree of physical knowledge is assumed. In ▶ System Identification: An Overview, various connections between physical models and parameter estimation are discussed.

Overview of Physical Modeling

Since modeling covers such a wide variety of physical systems, there are no universal systematic principles. However, a few concepts have wide application. One of them is the preservation of certain quantities like energy, leading to *balance equations.* A simple example is given by the heating of a body. If W is the energy stored as heat, P_1 an external power input, and P_2 the heat loss to the environment per time unit, energy balance gives

$$\frac{dW}{dt} = P_1 - P_2 \quad (2)$$

To get a complete model, one needs also *constitutive relations*, i.e., relations between relevant physical variables. For instance, one might know that the stored energy is proportional to the temperature T, $W = CT$ and that the energy loss is from black body radiation, $P_2 = kT^4$. The model is then
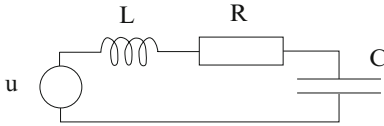
$$C\frac{dT}{dt} = P_1 - kT^4 \quad (3)$$

The model is now an ordinary differential equation with state variable T, input variable P_1 and parameters C and k.

Physical Analogies and General Structures

Physical Analogies

Physicists and engineers have noted that modeling in different areas of physics often gives very similar models. The term "analogies" is

M

Modeling of Dynamic Systems from First Principles, Fig. 1 Electric circuit

Modeling of Dynamic Systems from First Principles, Fig. 2 Mechanical system

often used in modeling to describe this fact. Here we will show some analogies between electrical and mechanical phenomena. Consider the electric circuit given in Fig. 1. An ideal voltage source is connected in series with an inductor, a resistor, and a capacitor. Using u and v to denote the voltages over the voltage source and capacitor, respectively, and i to denote the current, a mathematical model is

$$
C\frac{dv}{dt} = i
$$
$$
L\frac{di}{dt} + Ri + v = u \tag{4}
$$

The first equation uses the definition of capacitance and the second one uses Kirchhoff's voltage law. Compare this to the mechanical system of Fig. 2 where an external force F is applied to a mass m that is also connected to a damper b and a spring with spring constant k. If S is the elongation force of the spring and w the velocity of the mass, a system model is

$$
\frac{dS}{dt} = kw
$$
$$
m\frac{dw}{dt} + bw + S = F \tag{5}
$$

Here the first equation uses the definition of spring constant and the second one uses Newton's 2nd law. The models are seen to be the same with the following correspondences between time-varying quantities

$$
u \leftrightarrow F, \quad i \leftrightarrow w, \quad v \leftrightarrow S \tag{6}
$$

and between parameters

$$
C \leftrightarrow 1/k, \quad L \leftrightarrow m, \quad R \leftrightarrow b \tag{7}
$$

Note that the products (voltage) × (current) and (force) × (velocity) give the power.

Bond Graphs

The bond graph is a tool to do systematic modeling based on the analogies of the previous section. The basic element is the bond

$$
\overset{e}{\underset{f}{\longrightarrow}}
$$

formed by a half arrow showing the direction of positive energy flow. Two variables are associated with the bond, the *effort* variable e and the *flow* variable f. The product ef of these variables gives the power. In the electric domain e is voltage and f is current. For mechanical systems e is force, while f is velocity. Bond graph theory has three basic components to describe storage and dissipation of energy. The relations

$$
\alpha\frac{de}{dt} = f, \quad \beta\frac{df}{dt} = e, \quad \gamma f = e \tag{8}
$$

are known as C, I, and R elements, respectively. Input signals are modeled by elements called effort sources S_e or flow sources S_f, respectively. A bond graph describes the energy flow between these elements. When the energy flow is split, it can either be at s junctions where the flows are equal and the efforts are added or at a p junction where efforts are equal and flows are additive. The model (5), for instance, can be described by the bond graph in Fig. 3. The graph shows how the energy from the external force is split into the acceleration of the mass, the elongation of the spring, and dissipation into the damper. The splitting of the energy flow is accomplished by an s element, meaning that the velocity is the same for all elements but that the forces are added:
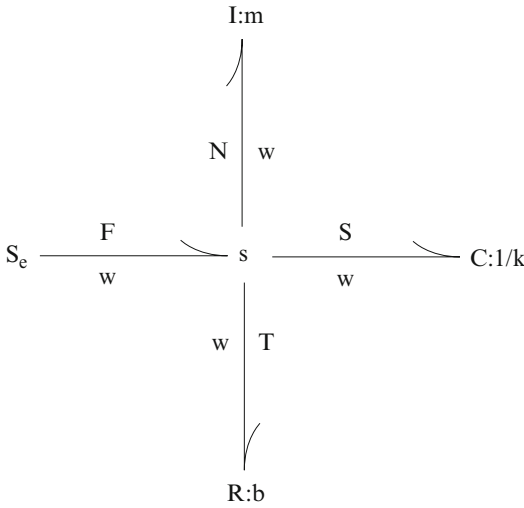
Modeling of Dynamic Systems from First Principles, Fig. 3 Bond graph for mechanical or electric system

$$F = N + S + T \qquad (9)$$

Here T and N denote the forces associated with the damper and the mass, respectively. From (8) it follows that

$$k^{-1}\frac{dS}{dt} = w, \quad m\frac{dw}{dt} = N, \quad bw = T \qquad (10)$$

Together (9) and (10) give the same model as (5). Using the correspondences (6), (7) it is seen that the same bond graph can also represent the electric circuit (4). An overview of bond graph modeling is given in Rosenberg and Karnopp (1983). A general overview of modeling, including bond graphs and the connection with identification, can be found in Ljung and Glad (1994b).

Port-Controlled Hamiltonian Systems

Many physical processes can be modeled as Hamiltonian systems. This means that there are state variables x, a scalar function H, and a skew symmetric matrix M so that the system dynamics is

$$\frac{dx}{dt} = M\nabla H(x) \qquad (11)$$

The function H is called the *Hamiltonian* of the system. To be useful in a control context, this model class has to be extended to handle inputs

and dissipation phenomena. To give an example the mechanical system used above is considered again.

Introduce x_1 as the length of the spring so that $dx_1/dt = w$. If H_1 is the energy stored in the spring, then the following relations hold:

$$H_1(x_1) = \frac{kx_1^2}{2}, \quad \frac{\partial H_1}{\partial x_1} = kx_1 = S \qquad (12)$$

Introducing x_2 for the momentum and H_2 as the kinetic energy, one has $dx_2/dt = N$ and

$$H_2(x_2) = \frac{x_2^2}{2m}, \quad \frac{\partial H_2}{\partial x_2} = m^{-1}x_2 = w \qquad (13)$$

Let $H = H_1 + H_2$ be the total energy. Then the following relation holds:

$$\frac{dx}{dt} = \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & b \end{bmatrix} \right) \begin{bmatrix} \partial H/\partial x_1 \\ \partial H/\partial x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \qquad (14)$$

This model is a special case of

$$\frac{dx}{dt} = (M - R)\nabla H(x) + Bu \qquad (15)$$

where M is a skew symmetric and R a nonnegative definite matrix, respectively. The model type is called a port-controlled Hamiltonian system with dissipation. Without external input ($B = 0$) and dissipation ($R = 0$), it reduces to an ordinary Hamiltonian system of the form (11). For systems generated by simple bond graphs, it can be shown that the junction structure gives the skew symmetric M, while the R elements give the matrix R. The storage of energy in I and C elements is reflected in H. The reader is directed to Duindam et al. (2009) for a description of the port Hamiltonian approach to modeling.

Component-Based Models and Modeling Languages

Since engineering systems are usually assembled from components, it is natural to treat their mathematical models in the same way. This is the idea behind block-oriented models where the output

M

of one model is connected to the input of another one:
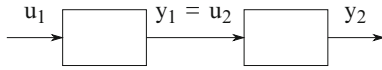
A nice feature of this block connection is that the state space description is preserved. Suppose the individual models are of the form (1)

$$\frac{dx_i}{dt} = f_i(x_i, u_i), \quad y_i = h_i(x_i, u_i), \quad i = 1, 2 \tag{16}$$

Then the connection $u_2 = y_1$ immediately gives the state space model

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f_1(x_1, u_1) \\ f_2(x_2, h_1(x_1, u_1)) \end{bmatrix}, \tag{17}$$
$$y_2 = h_2(x_2, h_1(x_1, u_1))$$

with input u_1, output y_2, and state $(x_1; x_2)$. This fact is the basis of block-oriented modeling and simulation tools like the MATLAB-based Simulink. Unfortunately the preservation of the state space structure does not extend to more general connections of systems. Consider, for instance, two pieces of rotating machinery described by

$$J_i \frac{d\omega_i}{dt} = -b_i\omega_i + M_i, \quad i = 1, 2 \tag{18}$$

where ω_i is the angular velocity, M_i the external torque, J_i the moment of inertia, and b_i the damping coefficient. Suppose the pieces are joined together so that they rotate with the same angular velocity. The mathematical model would then be

$$J_1 \frac{d\omega_1}{dt} = -b_1\omega_1 + M_1$$
$$J_2 \frac{d\omega_2}{dt} = -b_2\omega_2 + M_2 \tag{19}$$
$$\omega_1 = \omega_2$$
$$M_1 = -M_2$$

This is no longer a state space model of the form (1), but a mixture of dynamic and static relationships, usually referred to as a differential

algebraic equation (DAE). The difference from the connection of blocks in block diagrams is that now the connection is not between an input and an output. Instead there are the equations $\omega_1 = \omega_2$ and $M_1 = -M_2$ that destroy the state space structure. There exist modeling languages like Modelica (Fritzson 2000; Tiller 2001) or SimMechanics in MATLAB (MathWorks 2002) that accept this more general type of model. It is then possible to form model libraries of basic components that can be interconnected in very general ways to form models of complex systems. However, this more general structure poses some challenges when it comes to analysis and simulation that are described in the next section.

Differential Algebraic Equations (DAE)

This model (19) is a special case of the general differential algebraic equation

$$F(dz/dt, z, u) = 0 \tag{20}$$

A good description of both theory and numerical properties of such equations is given in Kunkel and Mehrmann (2006). In many cases it is possible to split the variables and equations in such a way that the following structure is achieved:

$$F(dz_1/dt, z_1, z_2, u) = 0, \quad F_2(z_1, z_2, u) = 0 \tag{21}$$

If z_2 can be solved from the second equation and substituted into the first one, and if dz_1/dt can then be solved from the first equation, the problem is reduced to an ordinary differential equation in z_1. Often, however, the situation is not as simple as that. For the example (19) an addition of the first two equations gives

$$(J_1 + J_2)\frac{d\omega_1}{dt} = -(b_1 + b_2)\omega_1 \tag{22}$$

which is a standard first-order system description. Note, however, that in order to arrive at this result, the relation $\omega_1 = \omega_2$ has to be differentiated. This DAE thus includes an implicit differentiation.

In the general case one can investigate how many times (20) has to be differentiated in order to get an explicit expression for dz/dt. This number is called the (differentiation) index. Both theoretical analysis and practical experience show that the numerical difficulties encountered when solving a DAE increase with increasing index; see, e.g., the classical reference Brenan et al. (1987). It turns out that mechanical systems in particular give high-index models when constructed by joining components, and this has been an obstacle to the use of DAE models. For linear DAE models the role of the index can be seen more easily. A linear model is given by

$$E\frac{dz}{dt} + Fz = Gu \qquad (23)$$

where the matrix E is singular (if E is invertible, multiplication with E^{-1} from the left gives an ordinary differential equation). The system can be transformed by multiplying with P from the left and changing variables with $z = Qw(P, Q$ nonsingular matrices). The transformed model is now

$$PEQ\frac{dw}{dt} + PFQw = PGu \qquad (24)$$

If $\lambda E + F$ is nonsingular for some value of the scalar λ, then it can be shown that there is a choice of P and Q such that (24) takes the form

$$\begin{bmatrix} I & 0 \\ 0 & N \end{bmatrix} \begin{bmatrix} dw_1/dt \\ dw_2/dt \end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \qquad (25)$$

where N is a nilpotent matrix, i.e., $N^k = 0$ for some positive integer k. The smallest such k turns out to be the index of the DAE. The transformed model (25) thus contains an ordinary differential equation:

$$\frac{dw_1}{dt} = Aw_1 + B_1u \qquad (26)$$

Using the nilpotency of N, the equation for w_2 can be rewritten:

$$w_2 = B_2u - NB_2\frac{du}{dt} + \cdots + (-N)^{k-1}B_2\frac{d^{k-1}u}{dt^{k-1}} \qquad (27)$$

This expression shows that an index $k > 1$ implies differentiation of the input (unless $N\,B_2$ happens to be zero). This in turn implies potential difficulties, e.g., if u is a measured signal.

Identification of DAE Models

The extended use of DAE models in modern modeling tools also means that there is a need to use these models in system identification. To fully use system identification theory, one needs a stochastic model of disturbances. The inclusion of such disturbances leads to a class of models described as stochastic differential algebraic equations. The treatment of such models leads to some interesting problems. In the previous section it was seen that DAE models often contain implicit differentiations of external signals. If a DAE model is to be well posed, this differentiation must not affect signals modeled as white noise. In Gerdin et al. (2007), conditions are given that guarantee that stochastic DAEs are well posed. There it is also described how a maximum likelihood estimate can be made for DAE models, laying the basis for parameter estimation.

Differential Algebra

For the case where models consist of polynomial equations, it is possible to manipulate them in a very systematic way. The model (20) is then generalized to

$$F(d^nz/dt^n, \ldots, dz/dt, z) = 0 \qquad (28)$$

where z is now a vector containing an arbitrary mix of inputs, outputs, and internal variables. There is then a theory based on Ritt (1950) that allows the transformation of (28) to a standard form where the properties of the system can be easily determined. The process is similar to the use of Gröbner bases but also includes the possibility of differentiating equations. Of particular interest to identification is the possibility of determining the identifiability of parameters with these tools. The model is then of the form

$$\begin{aligned} F(d^my/dt^m, \ldots, dy/dt, y, d^nz/dt^n, \\ \ldots, dz/dt, z; \theta) = 0 \end{aligned} \qquad (29)$$

where y contains measured signals, z contains unmeasured variables, and θ is a vector of parameters to be identified, while F is a vector of polynomials in these variables. It was shown in Ljung and Glad (1994a) that there is an algorithm giving for each parameter θ_k a polynomial:

$$g_k(d^m y/dt^m, \ldots, dy/dt, y; \; \theta_k) = 0 \quad (30)$$

This relation can be regarded as a polynomial in θ_k where all coefficients are expressed in measured quantities. The local or global identifiability will then be determined by the number of solutions. If θ_k is unidentifiable, then no equation of the form (30) will exist, and this fact will also be demonstrated by the output of the algorithm.

Summary and Future Directions

There is no general method to derive models from first principles. However, modeling techniques based on bond graphs or port-controlled Hamiltonian systems offer a systematic approach for large model classes. Modeling languages like Modelica make the practical work with modeling much easier. A fundamental problem that comes up is that models are not necessarily in state space form but are so called differential algebraic equation (DAE) models. Much of the future work is expected to deal with the handling of DAE models and in the development of modeling languages.

Cross-References

▶ Nonlinear System Identification: An Overview of Common Approaches
▶ System Identification: An Overview

Recommended Reading

A classical book on physical modeling is Rosenberg and Karnopp (1983) with emphasis on bond graph techniques. The physical modeling and identification perspectives are tied together in Ljung and Glad (1994b). A good reference for

Hamiltonian techniques is Duindam et al. (2009). The Modelica modeling language is treated in Tiller (2001) and Fritzson (2000). The former emphasizes the physical modeling point of view; the latter also gives details of the language itself.

Bibliography

Brenan KE, Campbell SL, Petzold LR (1987) Numerical solution of initial-value problems in differential-algebraic equations. Classics in applied mathematics (Book 14). SIAM, Philadelphia

Duindam V, Macchelli A, Stramigioli S, Bruyninckx H (eds) (2009) Modeling and control of complex physical systems: the Port-Hamiltonian approach. Springer, Berlin

Fritzson P (2000) Principles of object-oriented modeling and simulation with Modelica 2.1. IEEE/Wiley Interscience, Piscataway NJ

Gerdin M, Schön T, Glad T, Gustafsson F, Ljung L (2007) On parameter and state estimation for linear differential-algebraic equations. Automatica 43:416–425

Kunkel P, Mehrmann V (2006) Differential-algebraic equations. European Mathematical Society, Zürich

Ljung L, Glad ST (1994a) On global identifiability of arbitrary model parameterizations. Automatica 30(2):265–276

Ljung L, Glad T (1994b) Modeling of dynamic systems. Prentice Hall, Englewood Cliffs

Ritt JF (1950) Differential algebra. American Mathematical Society, Providence

Rosenberg RC, Karnopp D (1983) Introduction to physical system dynamics. McGraw-Hill, New York

The MathWorks (2002) SimMechanics. User's guide. The MathWorks, Natick

Tiller MM (2001) Introduction to physical modeling with Modelica. Kluwer Academic, Boston

Modeling, Analysis, and Control with Petri Nets

Manuel Silva
Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain

Abstract

Petri net is a generic term used to designate a broad family of related formalisms for discrete event views of (dynamic) Systems (DES), all

sharing some basic relevant features, such as *minimality* in the number of primitives, *locality* of the states and actions (with consequences for model construction), or *temporal realism*. The global state of a system is obtained by the juxtaposition of the different local states. We should initially distinguish between *autonomous* formalisms and those *extended by interpretation*. Models in the latter group are obtained by restricting the underlying autonomous behaviors by means of constraints that can be related to different kinds of external events, in particular to time. This article first describes *place/transition* nets (PT-nets), by default simply called Petri nets (PNs). Other formalisms are then mentioned. As a system theory modeling paradigm for concurrent DES, Petri nets are used in a wide variety of application fields.

Keywords

Condition/event nets (CE-nets); Continuous Petri nets (CPNs); Diagrams; Fluidization; Grafcet; Hybrid Petri nets (HPNs); Marking Petri nets; Place/transition nets (PT-nets); High-level Petri nets (HLPNs)

Introduction

Petri nets (PNs) are able to model concurrent and distributed DES (▶ Models for Discrete Event Systems: An Overview). They constitute a powerful family of formalisms with different expressive purposes and power. They may be applied to inter alia, modeling, logical analysis, performance evaluation, parametric optimization, dynamic control (minimum makespan, supervisory control, or other kinds), diagnosis, and implementation issues (eventually fault tolerant). Hybrid and continuous PNs are particularly useful when some parts of the system are highly populated. Being *multidisciplinary*, formalisms belonging to the Petri nets paradigm may cover several phases of the life cycle of complex DES.

A Petri net can be represented as a bipartite directed graph provided with arcs inscriptions; alternatively, this structure can be represented in algebraic form using some matrices. As in the case of differential equations, an initial condition or state should be defined in order to represent a dynamic system. This is done by means of an initial distributed state. The English translation of the Carl Adam Petri's seminal work, presented in 1962, is Petri (1966).

Untimed Place/Transition Net Systems

A place/transition net (PT-net) can be viewed as $\mathcal{N} = \langle P, T, \textbf{\textit{Pre}}, \textbf{\textit{Post}} \rangle$, where:

- P and T are disjoint and finite nonempty sets of *places* and *transitions*, respectively.
- **Pre** and **Post** are $|P| \times |T|$ sized, natural-valued (zero included), incidence matrices. The net is said to be *ordinary* if **Pre** and **Post** are valued on $\{0, 1\}$. Weighted arcs permit the abstract modeling of *bulk* services and arrivals.

A PT-net is a structure. The **Pre (Post)** function defines the connections from places to transitions (transitions to places). Those two functions can alternatively be defined as *weighted flow* relations (nets as graphs). Thus, PT-nets can be represented as *bipartite directed graphs* with *places* (p, using circles) and *transitions* (t, using bars or rectangles) as nodes: $\mathcal{N} = \langle P, T, F, W \rangle$, where $F \subseteq (P \times T) \cup (T \times P)$ is the *flow relation* (set of directed arcs, with $dom(F) \cup range(F) = P \cup T$), and $W : F \rightarrow \mathbb{N}^+$ assigns a natural weight to each arc.

The net structure represents the *static* part of the DES model. Furthermore, a "distributed state" is defined over the set of places, known as the *marking*. This is "numerically quantified" (not in an arbitrary alphabet, as in automata), associating *natural* values to the local state variables, the places. If a place p has a value $\upsilon(m(p) = \upsilon)$, it is said to have υ *tokens* (frequently depicted in graphic terms with υ black dots or just the number inside the place). The places are "state variables," while the markings are their "values"; the *global state* is defined through the concatenation of local states. The net structure, provided with an initial

M
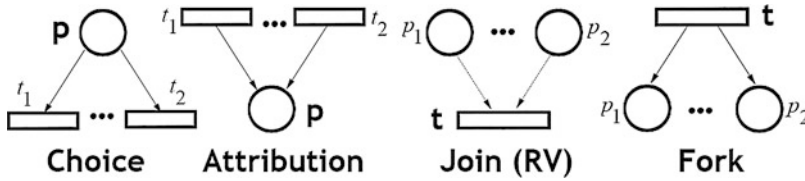
Modeling, Analysis, and Control with Petri Nets, Fig. 1 Most basic PN constructions: The logical OR is present around places, in *choices* (or branches) and *attributions* (or meets); the logical AND is formed around transitions, in *joins* (or waits or *rendezvous*) and *forks* (or splits)
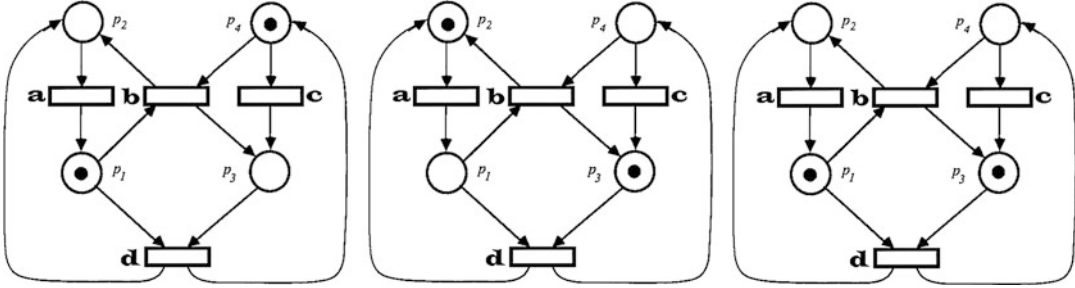
Modeling, Analysis, and Control with Petri Nets, Fig. 2 Only transitions *b* and *c* are initially enabled. The results of firing *b* or *c* are shown subsequently

marking, to be denoted as $(\mathcal{N}, m_0)$, is a *Petri net system*, or *marked Petri net*.

The last two basic PN constructions in Fig. 1 (*join* and *fork*) do not appear in finite-state machines; moreover, the arcs may be valued with natural numbers. The dynamic behavior of the net system (trajectories with changes in the marking) is produced by the firing of transitions, some "local operations" which follows very simple rules.

Markings in net systems evolve according to the following *firing* (or *occurrence*) rules (see, Fig. 2):

- A transition is said to be *enabled* at a given marking if each input place has at least as many tokens as the weight of the arc joining them.
- The *firing* or *occurrence* of an enabled transition is an instantaneous operation that removes from (adds to) each input (output) place a number of tokens equal to the weight of the arc joining the place (transition) to the transition (place).

The precondition of a transition can be seen as the resources required for the transition to be fired. The weight of the arc from a place to a transition represents the number of resources to be *consumed*. The post-condition defines the number resources *produced* by the firing of the transition. This is made explicit by the weights of the arcs from the transition to the places. Three important observations should be taken into account:

- The underlying logic in the firing of a transition is non-monotonic! It is a *consumption/production* logic.
- Enabled transitions are never *forced* to fire: This is a form of *non-determinism*.
- An *occurrence sequence* is a sequence of fired transitions $\sigma = t_1 \ldots t_k$. In the evolution from m_0, the reached marking m can be easily computed as:

$$m = m_0 + C \cdot \sigma, \qquad (1)$$

where $C = Post - Pre$ is the *token flow* matrix (*incidence* matrix if $\mathcal{N}$ is self-loop free) and σ the firing count vector corresponding to σ. Thus m and σ are vectors of natural numbers.

The previous equation is the *state-transition* equation (frequently known as the *fundamental* or, simply, *state* equation). Nevertheless, two important remarks should be made:

- It represents a necessary but not sufficient condition for reachability; the problem is that the existence of a $\boldsymbol{\sigma}$ does not guarantee that a corresponding sequence σ is firable from $\boldsymbol{m}_0$; thus, certain solutions – called *spurious* (Silva et al. 1998) – are not reachable. This implies that – except in certain net system subclasses – only semi-decision algorithms can usually be derived.
- All variables are natural numbers, which imply computational complexity.

It should be pointed out that in finite-state machines, the state is a single variable taking values in a symbolic unstructured set, while in PT-net systems, it is structured as a vector of nonnegative integers. This allows analysis techniques that do not require the enumeration of the state space.

At a structural level, observe that the negation is missing in Fig. 1; its inclusion leads to the so-called *inhibitor arcs*, an extension in expressive power. In its most basic form, if the place at the origin of an inhibitor arc is marked, it "inhibits" the enabling of the target transition. PT-net systems can model infinite-state systems, but not Turing machines. PT-net systems provided with inhibitor arcs (or *priorities* on the firing of transitions) can do it.

With this conceptually simple formalism, it is not difficult to express basic synchronization schemas (Fig. 3). All the illustrated examples use *joins*. When *weights* are allowed in the arcs, another kind of synchronization appears: Several copies of the same resource are needed (or produced) in a single operation. Being able to express *concurrency* and *synchronization*, when viewing the system at a higher level, it is possible to build *cooperation* and *competition* relationships.

Analysis and Control of Untimed PT Models

The behavior of a concurrent (eventually distributed) system is frequently difficult to understand and control. Thus, misunderstandings and mistakes are frequent during the design cycle. A way of cutting down the cost and duration of the

design process is to express in a formalized way properties that the system should enjoy and to use formal proof techniques. Errors can be eventually detected close to the moment they are introduced, reducing their propagation to subsequent stages. The goal in *verification* is to ensure that a given system is correct with respect to its specification (perhaps expressed in temporal-logic terms) or to a certain set of predetermined properties.

Among the most basic qualitative properties of "net systems" are the following: (1) *reachability* of a marking from a given one; (2) *boundedness*, characterizing finiteness of the state space; (3) *liveness*, related to potential fireability of all transitions starting on an arbitrary reachable marking (*deadlock-freeness* is a weaker condition in which only global infinite fireability of the net system model is guaranteed, even if some transitions no longer fire); (4) *reversibility*, characterizing recoverability of the initial marking from any reachable one; and (5) *mutual exclusion of two places*, dealing with the impossibility of reaching markings in which both places are simultaneously marked.

All the above are *behavioral* properties, which depend on the net system ($\mathcal{N}, \boldsymbol{m}_0$). In practice, sometimes problems with a net model are rooted in the net structure; thus, the study of the *structural* counterpart of certain behavioral properties may be of interest. For example, a "net" is *structurally bounded* if it is bounded for any initial marking; a "net" is *structurally live* if an initial marking exists that make the net system live (otherwise stated, it reflect non-liveness for arbitrary initial markings, a pathology of the net).

Basic techniques to analyze net systems include: (1) *enumeration*, in its most basic form based on the construction of a *reachability graph* (a sequentialized view of the behavior). If the net system is not bounded, losing some information, a finite *coverability graph* can be constructed; (2) *transformation*, based on an iterative rewriting process in which a net system enjoys a certain property if and only if a transformed ("simpler" to analyze) one also does. If the new system is easier to analyze, and the transformation is computationally cheap, the process may be extremely interesting; (3) *structural*, based on
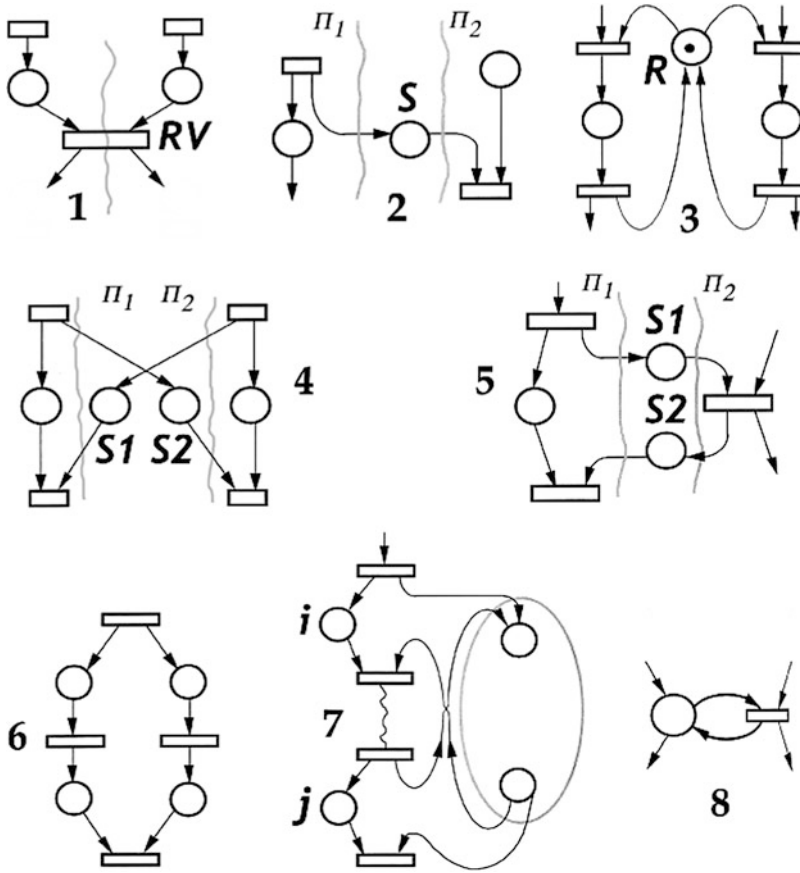
Modeling, Analysis, and Control with Petri Nets, Fig. 3 Basic synchronization schemes: (*1*) Join or *rendezvous, RV*; (*2*) *Semaphore, S*; (*3*) Mutual exclusion semaphore (*mutex*), R, representing a shared resource; (*4*) Symmetric RV built with two semaphores; (*5*) Asymmetric RV built with two semaphores (*master/slave*); (*6*) Fork-join (or *par-begin/par-end*); (*7*) Non-recursive *subprogram* (places *i* and *j* cannot be simultaneously marked – must be in mutex – to remember the returning point; for simplicity, it is assumed that the subprogram is single input/single output); (*8*) Guard (a self-loop from a place through a transition); its role is like a traffic light: If at least one token is present at the place, it allows the evolution, but it is not consumed. Synchronizations can also be modeled by the weights associated to the arcs going to transitions

graph properties, or in mathematical programming techniques rooted on the state equation; (4) *simulation*, particularly interesting for gaining certain confidence about the absence of certain pathological behaviors. Analysis strategies combining all these kinds of techniques are extremely useful in practice.

Reachability techniques provide sequentialized views for a particular initial marking. Moreover they suffer from the so-called *state explosion* problem. The reduction of this computational issue leads to techniques such as *stubborn set* methods (smaller, equally informative reachability graphs), also to non-sequentialized views such as those based on *unfoldings*. Transformation techniques are extremely useful, but not complete (i.e., not all net systems can be reduced in a practical way). In most cases, structural techniques only provide necessary or sufficient conditions (e.g., a sufficient condition for deadlock-freeness, a necessary condition for reversibility, etc.), but not a full characterization. As already pointed out, a limitation of methods based on the state equation for analyzing net systems is the existence of non reachable solutions (the so-called *spurious solutions*). In

this context, three kinds of related notions that must be differentiated are the following: (1) some natural vectors (left and right annullers of the token flow matrix, C: *P-semiflows* and *T-semiflows*), (2) some invariant laws (*token conservation* and *repetitive behaviors*), and (3) some peculiar subnets (*conservative* and *consistent* components, generated by the subsets of nodes in the P- and T-semiflows, respectively).

More than analysis, *control* leads to synthesis problems. The idea is *to enforce* the given system in order to fulfill a specification (e.g., to enforce certain mutual exclusion properties). Technically speaking, the idea is to "add" some elements in order to constrain the behavior in such a way that a correct execution is obtained. Questions related to control, observation, diagnosis, or identification are all areas of ongoing research.

With respect to classical control theory, there are two main differences: Models are DES and untimed (autonomous, fully nondeterministic, eventually labeling the transitions in order to be able to consider the PNs languages). Let us remark that for control purposes, the transitions should be partitioned into *controllable* (when enabled, you can either force or block the firing) and *uncontrollable* (if enabled, the firing is nondeterministic). A natural approach to synthesize a control is to start modeling the plant dynamics (by means of a PN, P) and adopting a specification for the desired closed-loop system (S). The goal is to compute a controller (L) such that S equals the parallel-composition of P and L; in other words, controllers (called "supervisors") are designed to ensure that only behaviors consistent with the specification may occur. The previous equality is not always possible, and the goal is usually relaxed to minimally limit the behavior within the specified legality (i.e., to compute *maximally permissive* controllers). For an approach in the framework of finite-state machines and regular languages, see ▸ Supervisory Control of Discrete-Event Systems. The synthesis in the framework of Petri nets and having goals as enforcing some generalized mutual exclusions constraints in markings or avoiding deadlocks, for example, can be efficiently approached by means of the

so named *structure theory*, based on the direct exploitation of the structure of the net model (using graph or mathematical programming theories and algorithms, where the initial marking is a parameter).

Similarly, transitions (or places) can be partitioned into *observable* and *unobservable*. Many *observability* problems may be of interest; for example, observing the firing of a subset of transitions to compute the subset of markings in which the system may be. Related to observability, *diagnosis* is the process of detecting a failure (any deviation of a system from its intended behavior) and identifying the cause of the abnormality. *Diagnosability*, like observability or controllability, is a logical criterion. If a model is diagnosable with respect to a certain subset of possible faults (i.e., it is possible to detect the occurrence of those faults in finite time), a diagnoser can be constructed (see section "Diagnosis and Diagnosability Analysis of Petri Nets" in ▸ Diagnosis of Discrete Event Systems). *Identification* of DES is also a question that has required attention in recent years. In general, the starting point is a behavioral observation, the goal being to construct a PN model that generates the observed behavior, either from examples/counterexamples of its language or from the structure of a reachability graph. So the results are *derived* models, not human-made models (i.e., not made by designers).

The Petri Nets Modeling Paradigm

Along the *life cycle* of DES, designers may deal with basic modeling, analysis, and synthesis from different perspectives together with implementation and operation issues. Thus, the designer may be interested in expressing the basic structure, understanding untimed possible behaviors, checking logic properties on the model when provided with some timing (e.g., in order to guarantee if a certain reaction is possible before 3 ms; something relevant in real-time systems), computing some performance indices on timed models (related to the throughput in the firing of a given transition, or to the length of a waiting line,

expressed as the number of tokens in a place), computing a schedule or control that optimizes a certain objective function, decomposing the model in order to prepare an efficient implementation, efficiently determining redundancies in order to increase the degree of fault tolerance, etc. For these different tasks, different formalisms may be used. Nevertheless, it seems desirable to have a *family* of related formalisms rather than a collection of "unrelated" or weakly related formalisms. The expected advantages would include *coherence* among models usable in different phases, *economy* in the transformations and *synergy* in the development of models and theories.

Other Untimed PN Formalisms: Levels of Expressive Power

PT-net systems are more powerful than *condition/event* (CE) systems, roughly speaking the basic seminal formalism of Carl Adam Petri in which places can be marked only with zero or one token (Boolean marking). CE-systems can model only finite-state systems. As already said, "extensions" of the expressive power of untimed PT-net systems to the level of Turing machines are obtained by adding inhibitor arcs or priorities to the firing of transitions.

An important idea is adding the notion of *individuals* to tokens (e.g., from anonymous to labeled or colored tokens). Information in tokens allows the objects to be named (they are no longer indistinguishable) and dynamic associations to be created. Moving from PT-nets to so-called *high-level PNs* (HLPNs) is something like "moving from assembler to high-level programming languages," or, at the computational level, like "moving from pure numerical to a symbolic level." There are many proposals in this respect, the more important being predicate/transition nets and colored PNs. Sometimes, this type of abstraction has the same theoretical expressiveness as PT-net systems (e.g., colored PNs if the number of colors is finite); in other words, high-level views may lead to more compact

and structured models, while keeping the same theoretical expressive power of PT-nets (i.e., we can speak of "abbreviations," not of "extensions"). In other cases, *object-oriented* concepts from computer programming are included in certain HLPNs. The analysis techniques of HLPNs can be approached with techniques based on enumeration, transformation, or structural considerations and simulation, generalizing those developed for PT-net systems.

Extending Net Systems with External Events and Time: Nonautonomous Formalisms

When dealing with net systems that interact with some specific environment, the marking evolution rule must be slightly modified. This can be done in an enormous number of ways, considering external events and logical conditions as inputs to the net model, in particular some depending on time. The same interpretation given to a graph in order to define a finite-*state diagram* can be used to define a *marking diagram*, a formalism in which the key point is to recognize that the state is now numerical (for PT-net systems) and distributed. For example, drawing a parallel with Moore automata, the transitions should be labeled with logical conditions and events, while unconditional actions are associated to the places. If a place is marked, the associated actions are emitted.

Even if only *time-based interpretations* are considered, there are a large number of successful proposals for formalisms. For example, it should be specified if time is associated to the firing of transitions (T-timing may be atomic or in three phases), to the residence of tokens in places (P-timed), to the arcs of the net, as tags to the tokens, etc. Moreover, even for T-timed models, there are many ways of defining the timing: *time intervals*, *stochastic* or *possibilistic* forms, and the *deterministic* case as a particular one. If the firing of transitions follows *exponential* pdfs, and the conflict resolution follows the *race* policy (i.e., fire in conflicts the first that ends the task,

not a *preselection* policy), the underlying Markov chain described is isomorphic to the reachability graph (due to the Markovian "memoryless" property). Moreover, the addition of *immediate* transitions (whose firing is instantaneous) enriches the practical modeling possibilities, eventually complicating the analysis techniques. Timed models are used to compute minimum and maximum time delays (when time intervals are provided, in real-time problems) or performance figures (throughput, utilization rate of resources, average number of tokens – clients – in services, etc.). For performance evaluation, there is an array of techniques to compute bounds, approximated values, or exact values, sometimes generalizing those that are used in certain queueing network classes of models. Simulation techniques are frequently very helpful in practice to produce an *educated guess* about the expected performance.

Time constraints on Petri nets may change logical properties of models (e.g., mutual exclusion constraints, deadlock-freeness, etc.), calling for new analysis techniques. For example, certain timings on transitions can transform a live system into a non-live one (if to the net system in Fig. 2 are associated deterministic times to transitions and a race policy with the time associated to transition *c* smaller than that of transition *a*, transition *b* cannot be fired, after firing transition *d*; thus it is non-live, while the untimed model was live). By the addition of some time constraints, the transformation of a non-live model into a live one is also possible. So additional analysis techniques need to be considered, redefining the state, now depending also on time, more than just on the marking.

Finally, as in any DES, the optimal control of timed Petri net models (scheduling, sequencing, etc.) may be approached by techniques as dynamic programming or perturbation analysis (presented in the context of queueing networks and Markov chains, see ▶ Perturbation Analysis of Discrete Event Systems). In practice, those problems are frequently approached by means of some partially heuristic strategies. About the diagnosis of timed Petri nets, see ▶ Diagnosis of Discrete Event Systems. Of course, all these tasks can be done with HLPNs.

Fluid and Hybrid PN Models

Different ideas may lead to different kinds of *hybrid* PNs. One is to *fluidize* (here to relax the natural numbers of discrete markings into the nonnegative reals) the firing of transitions that are "most time" enabled. Then the relaxed model has *discrete* and *continuous* transitions, thus also *discrete* and *continuous* places. If all transitions are fluidized, the PN system is said to be *fluid* or *continuous*, even if technically it is a hybrid one. In this approach, the main goal is to try to overcome the *state explosion* problem inherent to enumeration techniques. Proceeding in that way, some computationally NP-hard problems may become much easier to solve, eventually in polynomial time. In other words, *fluidization* is an abstraction that tries to make tractable certain real-scale DES problems (▶ Discrete Event Systems and Hybrid Systems, Connections Between).

When transitions are timed with the so-called *infinite server* semantics, the PN system can be observed as a time differentiable *piecewise affine system*. Thus, even if the relaxation "simplifies" computations, it should be taken into account that continuous PNs with infinite server semantics are able to simulate Turing machines. From a different perspective, the steady-state throughput of a given transition may be non-monotonic with respect to the firing rates or the initial marking (e.g., if faster or more machines are used, the uncontrolled system may be slower); moreover, due to the important expressive power, *discontinuities* may even appear with respect to continuous design parameters as firing rates, for example.

An alternative way to define hybrid Petri nets is a generalization of *hybrid automata*: The net system is a DES, but sets of differential equations are associated to the marking of places. If a place is marked, the corresponding differential equations contribute to define its evolution.

Summary and Future Directions

Petri nets designate a broad family of related DES formalisms (a modeling paradigm) each one specifically tailored to approach certain problems. Conceptual simplicity coexists with

powerful modeling, analysis, and synthesis capabilities. From a control theory perspective, much work remains to be done for both untimed and timed formalisms (remember, there are many different ways of timing), particularly when dealing with optimal control of timed models. In engineering practice, approaches to the latter class of problems frequently use heuristic strategies. From a broader perspective, future research directions include improvements required to deal with controllability and the design of controllers, with observability and the design of observers, with diagnosability and the design of diagnosers, and with identification. This work is not limited to the strict DES framework, but also applies to analogous problems relating to relaxations into *hybrid* or *fluid* approximations (particularly useful when high populations are considered). The distributed nature of system is more and more frequent and is introducing new constraints, a subject requiring serious attention. In all cases, different from firing languages approaches, the so named *structure theory* of Petri nets should gain more interest.

Cross-References

▶ Applications of Discrete-Event Systems
▶ Diagnosis of Discrete Event Systems
▶ Discrete Event Systems and Hybrid Systems, Connections Between
▶ Models for Discrete Event Systems: An Overview
▶ Perturbation Analysis of Discrete Event Systems
▶ Supervisory Control of Discrete-Event Systems

Recommended Reading

Topics related to PNs are considered in well over a hundred thousand papers and reports. The first generation of books concerning this field is Brauer (1980), immediately followed by Starke (1980), Peterson (1981), Brams (1983), Reisig (1985), and Silva (1985). The fact that they are written in English, French, German, and Spanish is proof of the rapid dissemination of this

knowledge. Most of these books deal essentially with PT-net systems. Complementary surveys are Murata (1989), Silva (1993), and David and Alla (1994), the latter also considering some continuous and hybrid models. Concerning high-level PNs, Jensen and Rozenberg (1991) is a selection of papers covering the main developments during the 1980s. Jensen and Kristensen (2009) focuses on state space methods and simulation where elements of timed models are taken into account, but performance evaluation of stochastic systems is not covered. Approaching the present day, relevant works written with complementary perspectives include inter alia, Girault and Valk (2003), Diaz (2009), David and Alla (2010), and Seatzu et al. (2013). The consideration of time in nets with an emphasis on performance and performability evaluation is addressed in monographs such as Ajmone Marsan et al. (1995), Bause and Kritzinger (1996), Balbo and Silva (1998), and Haas (2002), while timed models under different fuzzy interpretations are the subject of Cardoso and Camargo (1999). Structure-based approaches to controlling PN models is the main subject in Iordache and Antsaklis (2006) or Chen and Li (2013). Different kinds of hybrid PN models are studied in Di Febbraro et al. (2001), Villani et al. (2007), and David and Alla (2010), while a broad perspective about modeling, analysis, and control of continuous (untimed and timed) PNs is provided by Silva et al. (2011).

DiCesare et al. (1993) and Desrochers and Al-Jaar (1995) are devoted to the applications of PNs to manufacturing systems. A comprehensive updated introduction to business process systems and PNs can be found in van der Aalst and Stahl (2011). Special volumes dealing with other monographic topics are, for example, Billington et al. (1999), Agha et al. (2001), and Cortadella et al. (2002). An application domain for Petri nets emerging over the last two decades is *systems biology*, a model-based approach devoted to the analysis of biological systems (Koch et al. 2011; Wingender 2011). Furthermore, it should be pointed out that Petri nets have also been employed in many other application domains (e.g., from logistics to musical systems).

For an overall perspective of the field over the five decades that have elapsed since the publication of Carl Adam Petri's PhD thesis, including historical, epistemological, and technical aspects, see Silva (2013).

Bibliography

Agha G, de Cindio F, Rozenberg G (eds) (2001) Concurrent object-oriented programming and Petri nets, advances in Petri nets. Volume 2001 of LNCS. Springer, Berlin/Heidelberg/New York

Ajmone Marsan M, Balbo G, Conte G, Donatelli S, Franceschinis G (1995) Modelling with generalized stochastic Petri nets. Wiley, Chichester/New York

Balbo G, Silva M (eds) (1998) Performance models for discrete event systems with synchronizations: formalisms and analysis techniques. Proceedings of human capital and mobility MATCH performance advanced school, Jaca. Available online: http://webdiis.unizar.es/GISED/?q=news/matchbook

Bause F, Kritzinger P (1996) Stochastic Petri nets. an introduction to the theory. Vieweg, Braunschweig

Billington J, Diaz M, Rozenberg G (eds) (1999) Application of Petri nets to communication networks, advances in Petri nets. Volume 1605 of LNCS. Springer, Berlin/Heidelberg/New York

Brams GW (1983) Reseaux de Petri: Theorie et Pratique. Masson, Paris

Brauer W (ed) (1980) Net theory and applications. Volume 84 of LNCS. Springer, Berlin/New York

Cardoso J, Camargo H (eds) (1999) Fuzziness in Petri nets. Volume 22 of studies in fuzziness and soft computing. Physica-Verlag, Heidelberg/New York

Chen Y, Li Z (2013) Optimal supervisory control of automated manufacturing systems. CRC, Boca Raton

Cortadella J, Yakovlev A, Rozenberg G (eds) (2002) Concurrency and hardware design, advances in Petri nets. Volume 2549 of LNCS. Springer, Berlin/Heidelberg/New York

David R, Alla H (1994) Petri nets for modeling of dynamic systems – a survey. Automatica 30(2):175–202

David R, Alla H (2010) Discrete, continuous and hybrid Petri nets, Springer-Verlag, Berlin/Heidelberg

Desrochers A., Al-Jaar RY (1995) Applications of Petri nets in manufacturing systems. IEEE, New York

Diaz M (ed) (2009) Petri nets: fundamental models, verification and applications. Control systems, robotics and manufacturing series (CAM). Wiley, London

DiCesare F, Harhalakis G, Proth JM, Silva M, Vernadat FB (1993) Practice of Petri nets in manufacturing. Chapman & Hall, London/Glasgow/New York

Di Febbraro A, Giua A, Menga G (eds) (2001) Special issue on hybrid Petri nets. Discret Event Dyn Syst 11(1–2):5–185

Girault C, Valk R (2003) Petri nets for systems engineering. a guide to modeling, verification, and applications. Springer, Berlin

Haas PJ (2002) Stochastic Petri nets. modelling, stability, simulation. Springer series in operations research. Springer, New York

Iordache MV, Antsaklis PJ (2006) Supervisory control of concurrent systems: a Petri net structural approach. Birkhauser, Boston

Jensen K, Kristensen LM (2009) Coloured Petri nets. modelling and validation of concurrent systems. Springer, Berlin

Jensen K, Rozenberg G (eds) (1991) High-level Petri nets. Springer, Berlin

Koch I, Reisig W, Schreiber F (eds) (2011) Modeling in systems biology. the Petri net approach. Computational biology, vol 16. Springer, Berlin

Murata T (1989) Petri nets: properties, analysis and applications. Proc IEEE 77(4):541–580

Peterson JL (1981) Petri net theory and the modeling of systems. Prentice-Hall, Upper Saddle River

Petri CA (1966) Communication with automata. Rome Air Development Center-TR-65-377, New York

Reisig W (1985) Petri nets. an introduction. Volume 4 of EATCS monographs on theoretical computer science. Springer-Verlag, Berlin/Heidelberg/New York

Seatzu C, Silva M, Schuppen J (eds) (2013) Control of discrete-event systems. Automata and Petri net perspectives. Number 433 in lecture notes in control and information sciences. Springer, London

Silva M (1985) Las Redes de Petri: en la Automatica y la Informatica. Ed. AC, Madrid (2nd ed., Thomson-AC, 2002)

Silva M (1993) Introducing Petri nets. In: Practice of Petri nets in manufacturing. Chapman and Hall, London/New York, pp 1–62

Silva M (2013) Half a century after Carl Adam Petri's PhD thesis: a perspective on the field. Annu Rev Control 37(2):191–219

Silva M, Teruel E, Colom JM (1998) Linear algebraic and linear programming techniques for the analysis of net systems. Volume 1491 of LNCS, advances in Petri nets. Springer, Berlin/Heidelberg/New York, pp 309–373

Silva M, Julvez J, Mahulea C, Vazquez C (2011) On fluidization of discrete event models: observation and control of continuous Petri nets. Discret Event Dyn Syst 21:427–497

Starke P (1980) Petri-Netze. Deutcher Verlag der Wissenschaften, Berlin

van der Aalst W, Stahl C (2011) Modeling business processes: a Petri net oriented approach. MIT, Cambridge

Villani E, Miyagi PE, Valette R (2007) Modelling and analysis of hybrid supervisory systems. A Petri net approach. Springer, Berlin

Wingender E (ed) (2011) Biological Petri nets. Studies in health technology and informatics. vol 162. IOS Press, Lansdale

M

Model-Predictive Control in Practice

Thomas A. Badgwell[1] and S. Joe Qin[2]
[1]ExxonMobil Research & Engineering,
Annandale, NJ, USA
[2]University of Southern California, Los Angeles,
CA, USA

Synonyms

MPC

Abstract

This entry provides a brief description of model
predictive control (MPC) technology and how it
is used in practice. The emphasis here is on re-
fining and chemical plant applications where the
technology has achieved its greatest acceptance.
After a short description of what MPC is and
how it fits into the hierarchy of control functions,
the basic algorithm is presented as a sequence of
three optimization problems. The steps required
for a successful application are then outlined,
followed by a summary and outline of likely
future directions for MPC technology.

Keywords

Computer control; Mathematical programming;
Predictive control

Introduction

Model predictive control (MPC) refers to a class
of computer control algorithms that utilize an
explicit mathematical model to predict future
process behavior. At each control interval, in the
most general case, an MPC algorithm solves a
sequence of three nonlinear programs to answer
the following essential questions: where is the
process heading (state estimation), where should
the process go (steady-state target optimization),
and what is the best sequence of control (input)
adjustments to send it to the right place (dynamic
optimization). The first control (input) adjust-
ment is implemented and then the entire cal-
culation sequence is repeated at the subsequent
control cycles.

MPC technology arose first in the context of
petroleum refinery and power plant control prob-
lems (Cutler and Ramaker 1979; Richalet et al.
1978). Specific needs that drove the development
of MPC technology include the requirement for
economic optimization and strict enforcement
of safety and equipment constraints. Promising
early results led to a wave of successful industrial
applications, sparking the development of several
commercial offerings (Qin and Badgwell 2003)
and generating intense interest from the academic
community (Mayne et al. 2000). Today MPC
technology permeates the refining and chemical
industries and has gained increasing acceptance
in a wide variety of areas including chemicals,
automotive, aerospace, and food processing ap-
plications. The total number of MPC applications
worldwide was estimated in 2003 to be 4,500
(Qin and Badgwell 2003).

MPC Control Hierarchy

In a modern chemical plant or refinery, MPC
is part of a multilevel hierarchy, as illustrated
in Fig. 1. Moving from the top level to the
bottom, the control functions execute at a
higher frequency but cover a smaller geographic
scope. At the bottom level, referred to as
Level 0, proportional-integral-derivative (PID)
controllers execute several times a second within
distributed control system (DCS) hardware.
These controllers adjust individual valves to
maintain desired flows, pressures, levels, and
temperatures.

At Level 1, MPC runs once a minute
to perform dynamic constraint control for
an individual processing unit, such as crude
distillation unit or a fluid catalytic cracker (Gary
et al. 2007). It typically utilizes a linear dynamic
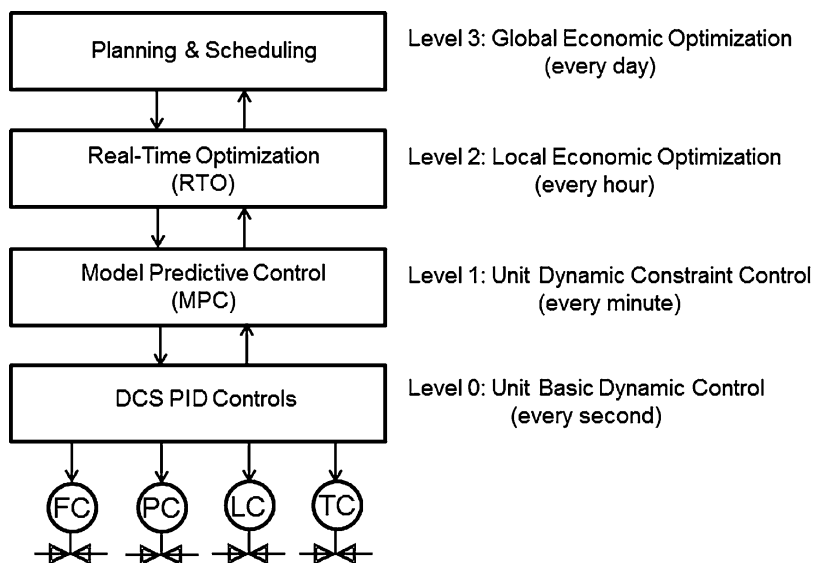
Model-Predictive Control in Practice, Fig. 1 Hierarchy of control functions in a refinery/chemical plant

model identified directly from process step-test data. The MPC has the job of holding the unit at the best economic operating point in the face of dynamic disturbances and operational constraints.

At Level 2, a real-time optimizer (RTO) runs hourly to calculate optimal steady-state targets for a collection of processing units. It uses a rigorous first-principles steady-state model to calculate targets for key operating variables such as unit temperatures and feed rates. These are typically passed down to several MPCs for implementation.

At Level 3, planning and scheduling functions are carried out daily to optimize economics for an entire chemical plant or refinery. Simple steady-state models are typically used at this level, with some nonlinear but mostly linear connections between model inputs and outputs. Key operating targets and economic data are typically passed to several RTO applications for implementation.

Note that a different mathematical model of the process is used at each level of the hierarchy. These models must be reconciled in some manner with current plant operation and with each other in order for the overall system to function properly.

MPC Algorithms

MPC algorithms function in much the same way that an experienced human operator would approach a control problem. Figure 2 illustrates the flow of information for a typical MPC implementation. At each control interval, the algorithm compares the current model output prediction y_p to the measured output y_m and passes the prediction error e and control (input) u to a state estimator, which estimates the dynamic state x. The most commonly used methods for state estimation can be viewed as special cases of an optimization-based formulation called moving horizon estimation (MHE) (Rawlings and Mayne 2009). The state estimate $\hat{x}$, which includes an estimate of the process disturbances $\hat{d}$, is then passed to a steady-state optimizer to determine the best operating point for the unit. The steady-state optimizer must also consider operator-entered output and control (input) targets y_t and u_t. The steady-state state and control (input) targets x_s and u_s are then passed, along with the state estimate $\hat{x}$, to a dynamic optimizer to compute the best trajectory of future control (input) adjustments. The first computed control (input) adjustment is then implemented and the entire calculation sequence is repeated at the
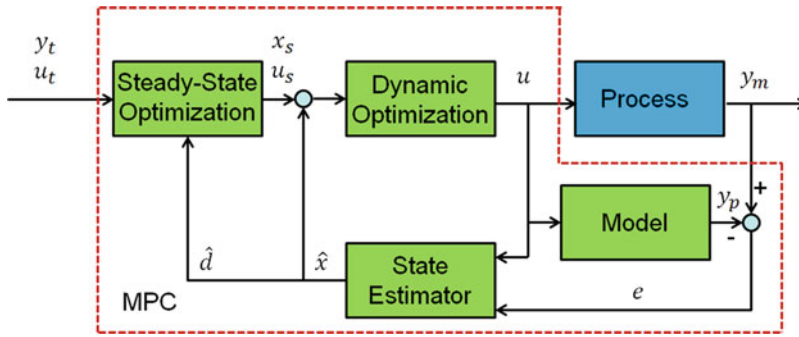
Model-Predictive Control in Practice, Fig. 2 Information flow for MPC algorithm

next control interval. The various commercial MPC algorithms differ in such details as the mathematical form of the dynamic model and the specific formulations of the state estimation, steady-state optimization, and dynamic optimization problems (Qin and Badgwell 2003).

In the general case, the MPC algorithm must solve the three optimization problems outlined above at each control interval. For the case of linear models and reasonable tuning parameters, these problems take the form of a convex quadratic program (QP) with a constant, positive-definite Hessian. As such, they can be solved relatively easily using standard optimization codes. For the case of a linear state-space model, the structure can be exploited even further to develop a specialized solution algorithm using an interior point method (Rao et al. 1998).

For the case of nonlinear models, these problems take the form of a nonlinear program (NLP) for which the solution domain is no longer convex, greatly complicating the numerical solution. A typical strategy is to iterate on a linearized version of the problem until convergence (Bielger 2010).

Implementation

The combined experience of thousands of MPC applications in the process industries has led to a near consensus on the steps required for a successful implementation:

- Justification – make the economic case for the application.

- Pre-test – design the control and test sensors and actuators.
- Step-test – generate process response data.
- Modeling – develop model from process response data.
- Configuration – configure the software and test preliminary tuning by simulation.
- Commissioning – turn on and test the controller.
- Post-audit – measure and certify economic performance.
- Sustainment – monitor and maintain the application.

The most expensive of these steps, both in terms of engineering time and lost production, is the generation of process response data through the step test. This is accomplished, in principle, by making significant adjustments to each variable that will be adjusted by the MPC while operating open loop to prevent compensating control action. This will necessarily cause abnormal movement in key operating variables, which may lead to lower throughput and off-spec products. Significant progress has been made in recent years to minimize these difficulties through the use of approximate closed-loop step testing (Darby and Nikolaou 2012).

Once the application has been commissioned, it is critical to set up an aggressive monitoring and sustainment program. MPC application benefits can fall off quickly due to changes in the process operation and as new personnel interact with it. New constraint variables may need to be added and key sections of the model may need

to be updated as time goes on. The mathematical problem of MPC monitoring remains a topic of current academic research (Zagrobelny et al. 2012).

Note that the implementation steps outlined above must be carried out by a carefully selected project team that typically includes, in addition to the MPC expert, an engineer with detailed knowledge of the process and an operator with significant relevant experience.

Summary and Future Directions

Model predictive control is now a mature technology in the process industries. A representative MPC algorithm in this domain includes a state estimator, a steady-state optimizer, and a dynamic optimizer, running once a minute. A successful MPC application usually starts with a careful economic justification, includes significant participation from process engineers and operators, and is maintained with an aggressive sustainment program. Many thousands of such applications are currently operating around the world, generating billions of dollars per year in economic benefits.

Likely future directions for MPC practice include increasing use of nonlinear models, improved state estimation through unmeasured disturbance modeling (Pannocchia and Rawlings 2003), and development of more efficient numerical solution methods (Zavala and Biegler 2009).

Cross-References

▶ Distributed Model Predictive Control
▶ Nominal Model-Predictive Control
▶ Optimization Algorithms for Model Predictive Control
▶ Tracking Model Predictive Control

Recommended Reading

The first descriptions of MPC technology appear in papers by Richalet et al. (1978) and Cutler

and Ramaker (1979). A detailed summary of the history of MPC technology development, as well as a survey of commercial offerings through 2003 can be found in the review article by Qin and Badgwell (2003). Darby and Nikolaou present a more recent summary of MPC practice (Darby and Nikolaou 2012). Textbook descriptions of MPC theory and design, suitable for classroom use, include Rawlings and Mayne (2009) and Maciejowski (2002). The book by Ljung (1999) provides a good summary of methods for identifying dynamic models from test data. Theoretical properties of MPC are analyzed in a highly cited paper by Mayne and coworkers (2000). Guidelines for designing disturbance models so as to achieve offset-free control can be found in Pannocchia and Rawlings (2003). Numerical solution strategies for the nonlinear programs found in MPC are discussed in the book by Biegler (2010). An efficient interior-point method for solving the linear MPC dynamic optimization is described in Rao et al. (1998). A promising algorithm for solving the nonlinear MPC dynamic optimization is outlined in Zavala and Biegler (2009). A data-based method for tuning Kalman Filters, which are often used for MPC state estimation, is described in Odelson et al. (2006). A new method for monitoring the performance of MPC is summarized in Zagrobelny et al. (2012). A readable summary of refining operations can be found in Gary et al. (2007).

Bibliography

Bielger LT (2010) Nonlinear programming, concepts, algorithms, and applications to chemical processes. SIAM, Philadelphia

Cutler CR, Ramaker BL (1979) Dynamic matrix control – a computer control algorithm. Paper presented at the AIChE national meeting, Houston, April 1979

Darby ML, Nikolaou M (2012) MPC: current practice and challenges. Control Eng Pract 20:328–342

Gary JH, Handwerk GE, Kaiser MJ (2007) Petroleum refining: technology and economics. CRC, New York

Ljung L (1999) System identification: theory for the user. Prentice Hall, Upper Saddle River

Maciejowski JM (2002) Predictive control with constraints. Pearson Education Limited, Essex

Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: stability and optimality. Automatica 36:789–814

M

Odelson BJ, Rajamani MR, Rawlings JB (2006) A new autocovariance least-squares method for estimating noise covariances. Automatica 42: 303–308

Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model predictive control. AIChE J 49:426–437

Qin SJ, Badgwell TA (2003) A survey of industrial model predictive control technology. Control Eng Pract 11:733–764

Rao CV, Wright SJ, Rawlings JB (1998) Application of interior-point methods to model predictive control. J Optim Theory Appl 99:723–757

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison

Richalet J, Rault A, Testud JL, Papon J (1978) Model predictive heuristic control: applications to industrial processes. Automatica 14:413–428

Zagrobelny M, Luo J, Rawlings JB (2012) Quis custodiet ipsos custodies? In: IFAC conference on nonlinear model predictive control 2012, Noordwijkerhout, Aug 2012

Zavala VM, Biegler LT (2009) The advanced-step NMPC controller: optimality, stability, and robustness. Automatica 45:86–93

Models for Discrete Event Systems: An Overview

Christos G. Cassandras
Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Synonyms

DES

Abstract

This article provides an introduction to discrete event systems (DES) as a class of dynamic systems with characteristics significantly distinguishing them from traditional time-driven systems. It also overviews the main modeling frameworks used to formally describe the operation of DES and to study problems related to their control and optimization.

Keywords

Introduction

Discrete event systems (DES) form an important class of dynamic systems. The term was introduced in the early 1980s to describe a DES in terms of its most critical feature: the fact that its behavior is governed by *discrete events* which occur asynchronously over time and which are solely responsible for generating state transitions. In between event occurrences, the state of a DES is unaffected. Examples of such behavior abound in technological environments, including computer and communication networks, manufacturing systems, transportation systems, logistics, and so forth. The operation of a DES is largely regulated by rules which are often unstructured and frequently human-made, as in initiating or terminating activities and scheduling the use of resources through controlled events (e.g., turning equipment "on"). On the other hand, their operation is also subject to uncontrolled randomly occurring events (e.g., a spontaneous equipment failure) which may or may not be observable through sensors. It is worth pointing out that the term "discrete event dynamic system" (DEDS) is also commonly used to emphasize the importance of the dynamical behavior of such systems (Cassandras and Lafortune 2008; Ho 1991).

There are two aspects of a DES that define its behavior:

1. The variables involved are both continuous and discrete, sometimes purely symbolic, i.e., nonnumeric (e.g., describing the state of a traffic light as "red" or "green"). This renders traditional mathematical models based on differential (or difference) equations inadequate and related methods based on calculus of limited use.

2. Because of the asynchronous nature of events that cause state transitions in a DES, it is neither natural nor efficient to use time as a synchronizing element driving its dynamics.

It is for this reason that DES are often referred to as *event driven*, to contrast them to classical *time-driven* systems based on the laws of physics; in the latter, as time evolves state, variables such as position, velocity, temperature, voltage, etc., also continuously evolve. In order to capture event-driven state dynamics, however, different mathematical models are necessary.

In addition, uncertainties are inherent in the technological environments where DES are encountered. Therefore, associated mathematical models and methods for analysis and control must incorporate such uncertainties. Finally, complexity is also inherent in DES of practical interest, usually manifesting itself in the form of combinatorially explosive state spaces. Although purely analytical methods for DES design, analysis, and control are limited, they have still enabled reliable approximations of their dynamic behavior and the derivation of useful structural properties and provable performance guarantees. Much of the progress made in this field, however, has relied on new paradigms characterized by a combination of mathematical techniques, computer-based tools, and effective processing of experimental data.

Event-driven and time-driven system components are often viewed as coexisting and interacting and are referred to as *hybrid systems* (separately considered in the Encyclopedia, including the article ▶ Discrete Event Systems and Hybrid Systems, Connections Between). Arguably, most contemporary technological systems are combinations of time-driven components (typically, the physical parts of a system) and event-driven components (usually, the computer-based controllers that collect data from and issue commands to the physical parts).

Event-Driven vs. Time-Driven Systems

In order to explain the difference between time-driven and event-driven behavior, we begin with the concept of "event." An event should be thought of as occurring instantaneously and causing transitions from one system state value to another. It may be identified with an action (e.g., pressing a button), a spontaneous natural occurrence (e.g., a random equipment failure), or the result of conditions met by the system state (e.g., the fluid level in a tank exceeds a given value). For the purpose of developing a model for DES, we will use the symbol e to denote an event. Since a system is generally affected by different types of events, we assume that we can define a discrete *event set* E with $e \in E$.

In a classical system model, the "clock" is what drives a typical state trajectory: with every "clock tick" (which may be thought of as an "event"), the state is expected to change, since continuous state variables continuously change with time. This leads to the term *time driven*. In the case of time-driven systems described by continuous variables, the field of systems and control has based much of its success on the use of well-known differential-equation-based models, such as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (1)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (2)$$

where (1) is a (vector) state equation with initial conditions specified and (2) is a (vector) output equation. As is common in system theory, $\mathbf{x}(t)$ denotes the state of the system, $\mathbf{y}(t)$ is the output, and $\mathbf{u}(t)$ represents the input, often associated with controllable variables used to manipulate the state so as to attain a desired output. Common physical quantities such as position, velocity, temperature, pressure, flow, etc., define state variables in (1). The state generally changes as time changes, and, as a result, the time variable t (or some integer $k = 0, 1, 2, \ldots$ in discrete time) is a natural independent variable for modeling such systems.

In contrast, in a DES, time no longer serves the purpose of driving such a system and may no longer be an appropriate independent variable. Instead, at least some of the state variables are discrete, and their values change only at certain points in time through instantaneous transitions which we associate with "events." If a clock is used, consider two possibilities: (i) At every

M

clock tick, an event e is selected from the event set E (if no event takes place, we use a "null event" as a member of E such that it causes no state change), and (ii) at various time instants (not necessarily known in advance or coinciding with clock ticks), some event e "announces" that it is occurring. Observe that in (i) state transitions are *synchronized* by the clock which is solely responsible for any possible state transition. In (ii), every event $e \in E$ defines a distinct process through which the time instants when e occurs are determined. State transitions are the result of combining these *asynchronous* concurrent event processes. Moreover, these processes need not be independent of each other. The distinction between (i) and (ii) gives rise to the terms *time-driven* and *event-driven* systems, respectively.

Comparing state trajectories of time-driven and event-driven systems is useful in understanding the differences between the two and setting the stage for DES modeling frameworks. Thus, in Fig. 1, we observe the following: (i) For the time-

driven system shown, the state space X is the set of real numbers $\mathbb{R}$, and $x(t)$ can take any value from this set. The function $x(t)$ is the solution of a differential equation of the general form $\dot{x}(t) = f(x(t), u(t), t)$, where $u(t)$ is the input. (ii) For the event-driven system, the state space is some discrete set $X = \{s_1, s_2, s_3, s_4\}$. The sample path can only jump from one state to another whenever an event occurs. Note that an event may take place, but not cause a state transition, as in the case of e_4. There is no immediately obvious analog to $\dot{x}(t) = f(x(t), u(t), t)$, i.e., no mechanism to specify how events might interact over time or how their time of occurrence might be determined. Thus, a large part of the early developments in the DES field has been devoted to the specification of an appropriate mathematical model containing the same expressive power as (1)–(2) (Baccelli et al. 1992; Cassandras and Lafortune 2008; Glasserman and Yao 1994).

We should point out that a time-driven system with continuous state variables, usually modeled through (1)–(2), may be abstracted as a DES through some form of discretization in time and quantization in the state space. We should also point out that discrete *event* systems should not be confused with discrete *time* systems. The class of discrete time systems contains both time-driven and event-driven systems.
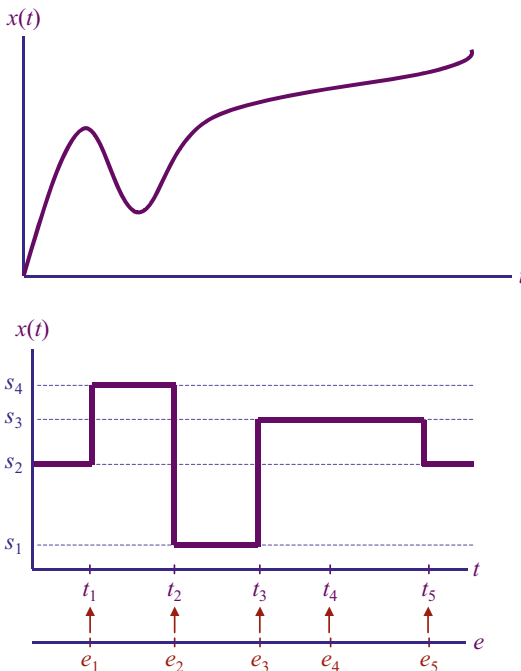
Models for Discrete Event Systems: An Overview, Fig. 1 Comparison of time-driven and event-driven state trajectories

Timed and Untimed Models of Discrete Event Systems

Returning to Fig. 1, instead of constructing the piecewise constant function $x(t)$ as shown, it is convenient to simply write the timed sequence of events $\{(e_1, t_1), (e_2, t_2), (e_3, t_3), (e_4, t_4), (e_5, t_5)\}$ which contains the same information as the state trajectory. Assuming that the initial state of the system (s_2 in this case) is known and that the system is "deterministic" in the sense that the next state after the occurrence of an event is unique, we can recover the state of the system at any point in time and reconstruct the DES state trajectory. The set of all possible timed sequences of events that a given system can ever

execute is called the *timed language* model of the system. The word "language" comes from the fact that we can think of the event E as an "alphabet" and of (finite) sequences of events as "words" (Hopcroft and Ullman 1979). We can further refine such a model by adding statistical information regarding the set of state trajectories (sample paths) of the system. Let us assume that probability distribution functions are available about the "lifetime" of each event type $e \in E$, that is, the elapsed time between successive occurrences of this particular e. A *stochastic timed language* is a timed language together with associated probability distribution functions for the events.

Stochastic timed language modeling is the most detailed in the sense that it contains event information in the form of event occurrences and their orderings, information about the exact times at which the events occur (not only their relative ordering), and statistical information about successive occurrences of events. If we delete the timing information from a timed language, we obtain an *untimed language*, or simply *language*, which is the set of all possible orderings of events that could happen in the given system. For example, the untimed sequence corresponding to the timed sequence of events in Fig. 1 is $\{e_1, e_2, e_3, e_4, e_5\}$.

Untimed and timed languages represent different levels of abstraction at which DES are modeled and studied. The choice of the appropriate level of abstraction clearly depends on the objectives of the analysis. In many instances, we are interested in the "logical behavior" of the system, that is, in ensuring that all the event sequences it can generate satisfy a given set of specifications, e.g., maintaining a precise ordering of events. In this context, the actual timing of events is not required, and it is sufficient to model only the untimed behavior of the system. *Supervisory control* that is discussed in the article ▸ Supervisory Control of Discrete-Event Systems is the term established for describing the systematic means (i.e., enabling or disabling events which are controllable) by which the logical behavior of a DES is regulated to achieve a given

specification (Cassandras and Lafortune 2008; Moody and Antsaklis 1998; Ramadge and Wonham 1987).

On the other hand, we may be interested in *event timing* in order to answer questions such as the following: "How much time does the system spend at a particular state?" or "Can this sequence of events be completed by a particular deadline?" More generally, event timing is important in assessing the performance of a DES often measured through quantities such as *throughput* or *response time*. In these instances, we need to consider the timed language model of the system. Since DES frequently operate in a stochastic setting, an additional level of complexity is introduced, necessitating the development of probabilistic models and related analytical methodologies for design and performance analysis based on stochastic timed language models. *Sample path analysis* and *perturbation analysis*, discussed in the entry ▸ Perturbation Analysis of Discrete Event Systems, refer to the study of sample paths of DES, focusing on the extraction of information for the purpose of efficiently estimating performance sensitivities of the system and, ultimately, achieving online control and optimization (Cassandras and Lafortune 2008; Glasserman 1991; Ho and Cao 1991; Ho and Cassandras 1983).

These different levels of abstraction are complementary, as they address different issues about the behavior of a DES. Although the language-based approach to DES modeling is attractive, it is by itself not convenient to address verification, controller synthesis, or performance issues. This motivates the development of *discrete event modeling formalisms* which represent languages in a manner that highlights structural information about the system behavior and can be used to address analysis and controller synthesis issues. Next, we provide an overview of three major modeling formalisms which are used by most (but not all) system and control theoretic methodologies pertaining to DES. Additional modeling formalisms encountered in the computer science literature include process algebras (Baeten and Weijland 1990) and communicating sequential processes (Hoare 1985).

M

Automata

A *deterministic automaton*, denoted by G, is a six-tuple

$$G = (\mathcal{X}, \mathcal{E}, f, \Gamma, x_0, \mathcal{X}_m),$$

where $\mathcal{X}$ is the set of *states*, $\mathcal{E}$ is the finite set of *events* associated with the transitions in G, and $f : \mathcal{X} \times \mathcal{E} \rightarrow \mathcal{X}$ is the *transition function*; specifically, $f(x, e) = y$ means that there is a transition labeled by event e from state x to state y and, in general, f is a *partial* function on its domain. $\Gamma : \mathcal{X} \rightarrow 2^{\mathcal{E}}$ is the *active event function* (or feasible event function); $\Gamma(x)$ is the set of all events e for which $f(x, e)$ is defined and it is called the *active event set* (or feasible event set) of G at x. Finally, x_0 is the *initial state* and $\mathcal{X}_m \subseteq \mathcal{X}$ is the set of *marked states*. The terms *state machine* and *generator* (which explains the notation G) are also used to describe the above object. Moreover, if $\mathcal{X}$ is a finite set, we call G a *deterministic finite-state automaton*. A *nondeterministic* automaton is defined by means of a relation over $\mathcal{X} \times \mathcal{E} \times \mathcal{X}$ or, equivalently, a function from $\mathcal{X} \times \mathcal{E}$ to $2^{\mathcal{X}}$.

The automaton G operates as follows. It starts in the initial state x_0, and upon the occurrence of an event $e \in \Gamma(x_0) \subseteq \mathcal{E}$, it makes a transition to state $f(x_0, e) \in \mathcal{X}$. This process then continues based on the transitions for which f is defined. Note that an event may occur without changing the state, i.e., $f(x, e) = x$. It is also possible that two distinct events occur at a given state causing the exact same transition, i.e., for $a, b \in \mathcal{E}$, $f(x, a) = f(x, b) = y$. What is interesting about the latter fact is that we may not be able to distinguish between events a and b by simply observing a transition from state x to state y.

For the sake of convenience, f is always extended from domain $\mathcal{X} \times \mathcal{E}$ to domain $\mathcal{X} \times \mathcal{E}^*$, where $\mathcal{E}^*$ is the set of *all* finite strings of elements of $\mathcal{E}$, including the empty string (denoted by ε); the $*$ operation is called the *Kleene closure*. This is accomplished in the following recursive manner: $f(x, \varepsilon) := x$ and $f(x, se) := f(f(x, s), e)$ for $s \in \mathcal{E}^*$ and $e \in \mathcal{E}$. The (untimed) language generated by G and denoted by $\mathcal{L}(G)$ is the set of all strings in $\mathcal{E}^*$ for which the extended function f is defined. The automaton model above is one instance of what is referred to as a *generalized semi-Markov scheme* (GSMS) in the literature of stochastic processes. A GSMS is viewed as the basis for extending automata to incorporate an event timing structure as well as nondeterministic state transition mechanisms, ultimately leading to *stochastic timed automata*, discussed in the sequel.

Let us allow for generally countable sets $\mathcal{X}$ and E and leave out of the definition any consideration for marked states. Thus, we begin with an automaton model $(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0)$. We extend the modeling setting to *timed* automata by incorporating a "clock structure" associated with the event set E which now becomes the input from which a specific event sequence can be deduced. The *clock structure* (or *timing structure*) associated with an event set $\mathcal{E}$ is a set $\mathbf{V} = \{\mathbf{v}_i : i \in \mathcal{E}\}$ of clock (or lifetime) sequences

$$\mathbf{v}_i = \{v_{i,1}, v_{i,2}, \ldots\}, i \in \mathcal{E}, v_{i,k} \in \mathbb{R}^+, k = 1, 2, \ldots$$

Timed Automaton. A *timed automaton* is defined as a six-tuple

$$(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0, \mathbf{V}),$$

where $\mathbf{V} = \{\mathbf{v}_i : i \in \mathcal{E}\}$ is a clock structure and $(\mathcal{X}, \mathcal{E}, f, \Gamma, x_0)$ is an automaton. The automaton generates a state sequence $x' = f(x, e')$ driven by an event sequence $\{e_1, e_2, \ldots\}$ generated through

$$e' = \arg \min_{i \in \Gamma(x)} \{y_i\} \tag{3}$$

with the clock values $y_i, i \in \mathcal{E}$, defined by

$$y_i' = \begin{cases} y_i - y^* & \text{if } i \neq e' \text{ and } i \in \Gamma(x) \\ v_{i, N_i + 1} & \text{if } i = e' \text{ or } i \notin \Gamma(x) \end{cases} \quad i \in \Gamma(x') \tag{4}$$

where the *interevent time* y^* is defined as

$$y^* = \min_{i \in \Gamma(x)} \{y_i\} \tag{5}$$

and the *event scores* N_i, $i \in \mathcal{E}$, are defined by

$$N_i' = \begin{cases} N_i + 1 \text{ if } i = e' \text{ or } i \notin \Gamma(x) \\ N_i \qquad \text{otherwise} \end{cases} \quad i \in \Gamma(x'). \tag{6}$$

In addition, initial conditions are $y_i = v_{i,1}$ and $N_i = 1$ for all $i \in \Gamma(x_0)$. If $i \notin \Gamma(x_0)$, then y_i is undefined and $N_i = 0$.

A simple interpretation of this elaborate definition is as follows. Given that the system is at some state x, the next event e' is the one with the smallest clock value among all feasible events $i \in \Gamma(x)$. The corresponding clock value, y^*, is the interevent time between the occurrence of e and e', and it provides the amount by which the time, t, moves forward: $t' = t + y^*$. Clock values for all events that remain active in state x' are decremented by y^*, except for the triggering event e' and all newly activated events, which are assigned a new lifetime v_{i,N_i+1}. Event scores are incremented whenever a new lifetime is assigned to them. It is important to note that the "system clock" t is fully controlled by the occurrence of events, which cause it to move forward; if no event occurs, the system remains at the last state observed.

Comparing $x' = f(x, e')$ to the state equation (1) for time-driven systems, we see that the former can be viewed as the event-driven analog of the latter. However, the simplicity of $x' = f(x, e')$ is deceptive: unless an event sequence is given, determining the *triggering* event e' which is required to obtain the next state x' involves the combination of (3)–(6). Therefore, the analog of (1) as a "canonical" state equation for a DES requires all Eqs. (3)–(6). Observe that this timed automaton generates a timed language, thus extending the untimed language generated by the original automaton G.

In the definition above, the clock structure **V** is assumed to be fully specified in a deterministic sense and so are state transitions dictated by $x' = f(x, e')$. The sequences $\{v_i\}$, $i \in \mathcal{E}$, can be extended to be specified only as stochastic sequences through distribution functions denoted by F_i, $i \in \mathcal{E}$. Thus, the *stochastic clock structure* (or *stochastic timing structure*) associated with

an event set $\mathcal{E}$ is a set of distribution functions $F = \{F_i : i \in \mathcal{E}\}$ characterizing the stochastic clock sequences

$$\{V_{i,k}\} = \{V_{i,1}, V_{i,2}, \ldots\}, \ i \in \mathcal{E},$$
$$V_{i,k} \in \mathbb{R}^+, \ k = 1, 2, \ldots$$

It is usually assumed that each clock sequence consists of random variables which are independent and identically distributed (iid) and that all clock sequences are mutually independent. Thus, each $\{V_{i,k}\}$ is completely characterized by a distribution function $F_i(t) = P[V_i \leq t]$. There are, however, several ways in which a clock structure can be extended to include situations where elements of a sequence $\{V_{i,k}\}$ are correlated or two clock sequences are interdependent. As for state transitions which may be nondeterministic in nature, such behavior is modeled through state transition probabilities as explained next.

Stochastic Timed Automaton. We can extend the definition of a timed automaton by viewing the state, event, and all event scores and clock values as random variables denoted respectively by capital letters X, E, N_i, and Y_i, $i \in \mathcal{E}$. Thus, a *stochastic timed automaton* is a six-tuple

$$(\mathcal{X}, \mathcal{E}, \Gamma, p, p_0, F),$$

where $\mathcal{X}$ is a countable *state space*; $\mathcal{E}$ is a countable *event set*; $\Gamma(x)$ is the *active event set* (or feasible event set); $p(x'; x, e')$ is a *state transition probability* defined for all $x, x' \in \mathcal{X}$, $e' \in \mathcal{E}$ and such that $p(x'; x, e') = 0$ for all $e' \notin \Gamma(x)$; p_0 is the probability mass function $P[X_0 = x]$, $x \in \mathcal{X}$, of the initial state X_0; and F is a *stochastic clock structure*. The automaton generates a stochastic state sequence $\{X_0, X_1, \ldots\}$ through a transition mechanism (based on observations $X = x$, $E' = e'$):

$$X' = x' \text{ with probability } p(x'; x, e') \quad (7)$$

and it is driven by a stochastic event sequence $\{E_1, E_2, \ldots\}$ generated exactly as in (3)–(6) with random variables E, Y_i, and N_i, $i \in \mathcal{E}$, instead

of deterministic quantities and with $\{V_{i,k}\} \sim F_i$ ($\sim$ denotes "with distribution"). In addition, initial conditions are $X_0 \sim p_0(x)$, $Y_i = V_{i,1}$, and $N_i = 1$ if $i \in \Gamma(X_0)$. If $i \notin \Gamma(X_0)$, then Y_i is undefined and $N_i = 0$.

It is conceivable for two events to occur at the same time, in which case we need a priority scheme to overcome a possible ambiguity in the selection of the triggering event in (3). In practice, it is common to expect that every F_i in the clock structure is absolutely continuous over $[0, \infty)$ (so that its density function exists) and has a finite mean. This implies that two events can occur at exactly the same time only with probability 0.

A stochastic process $\{X(t)\}$ with state space $\mathcal{X}$ which is generated by a stochastic timed automaton $(\mathcal{X}, \mathcal{E}, \Gamma, p, p_0, F)$ is referred to as a *generalized semi-Markov process* (GSMP). This process is used as the basis of much of the sample path analysis methods for DES (see Cassandras and Lafortune 2008; Glasserman 1991; Ho and Cao 1991).

Petri Nets

An alternative modeling formalism for DES is provided by *Petri nets*, originating in the work of C. A. Petri in the early 1960s. Like an automaton, a Petri net (Peterson 1981) is a device that manipulates events according to certain rules. One of its features is the inclusion of explicit conditions under which an event can be enabled. The Petri net modeling framework is the subject of the article ▶ Modeling, Analysis, and Control with Petri Nets. Thus, we limit ourselves here to a brief introduction. First, we define a Petri net graph, also called the *Petri net structure*. Then, we adjoin to this graph an initial state, a set of marked states, and a transition labeling function, resulting in the complete Petri net model, its associated dynamics, and the languages that it generates and marks.

Petri Net Graph. A Petri net is a directed *bipartite graph* with two types of nodes, *places* and *transitions*, and arcs connecting them. Events

are associated with transition nodes. In order for a transition to occur, several conditions may have to be satisfied. Information related to these conditions is contained in place nodes. Some such places are viewed as the "input" to a transition; they are associated with the conditions required for this transition to occur. Other places are viewed as the output of a transition; they are associated with conditions that are affected by the occurrence of this transition. A *Petri net graph* is formally defined as a weighted directed bipartite graph (P, T, A, w) where P is the finite set of *places* (one type of node in the graph), T is the finite set of *transitions* (the other type of node in the graph), $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs with directions from places to transitions and from transitions to places in the graph, and $w : A \rightarrow \{1, 2, 3, \ldots\}$ is the *weight function* on the arcs. Let $P = \{p_1, p_2, \ldots, p_n\}$, and $T = \{t_1, t_2, \ldots, t_m\}$. It is convenient to use $I(t_j)$ to represent the set of input places to transition t_j. Similarly, $O(t_j)$ represents the set of output places from transition t_j. Thus, we have $I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}$ and $O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}$.

Petri Net Dynamics. *Tokens* are assigned to places in a Petri net graph in order to indicate the fact that the condition described by that place is satisfied. The way in which tokens are assigned to a Petri net graph defines a *marking*. Formally, a *marking* x of a Petri net graph (P, T, A, w) is a function $x : P \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}$. Marking x defines row vector $\mathbf{x} = [x(p_1), x(p_2), \ldots, x(p_n)]$, where n is the number of places in the Petri net. The ith entry of this vector indicates the (nonnegative integer) number of tokens in place p_i, $x(p_i) \in \mathbb{N}$. In Petri net graphs, a token is indicated by a dark dot positioned in the appropriate place. The *state* of a Petri net is defined to be its marking vector $\mathbf{x}$. The state transition mechanism of a Petri net is captured by the structure of its graph and by "moving" tokens from one place to another. A transition $t_j \in T$ in a Petri net is said to be *enabled* if

$$x(p_i) \geq w(p_i, t_j) \text{ for all } p_i \in I(t_j) . \quad (8)$$

In words, transition t_j in the Petri net is enabled when the number of tokens in p_i is at least as large as the weight of the arc connecting p_i to t_j, for all places p_i that are input to transition t_j. When a transition is enabled, it can occur or *fire*. The *state transition function* of a Petri net is defined through the change in the state of the Petri net due to the firing of an enabled transition. The state transition function, $f : \mathbb{N}^n \times T \to \mathbb{N}^n$, of Petri net (P, T, A, w, x) is defined for transition $t_j \in T$ if and only if (8) holds. Then, we set $\mathbf{x}' = f(\mathbf{x}, t_j)$ where

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i),$$
$$i = 1, \ldots, n . \tag{9}$$

An "enabled transition" is therefore equivalent to a "feasible event" in an automaton. But whereas in automata the state transition function enumerates all feasible state transitions, here the state transition function is based on the structure of the Petri net. Thus, the next state defined by (9) explicitly depends on the input and output places of a transition and on the weights of the arcs connecting these places to the transition. According to (9), if p_i is an input place of t_j, it loses as many tokens as the weight of the arc from p_i to t_j; if it is an output place of t_j, it gains as many tokens as the weight of the arc from t_j to p_i. Clearly, it is possible that p_i is both an input and output place of t_j.

In general, it is entirely possible that, after several transition firings, the resulting state is $\mathbf{x} = [0, \ldots, 0]$ or that the number of tokens in one or more places grows arbitrarily large after an arbitrarily large number of transition firings. The latter phenomenon is a key difference with automata, where finite-state automata have only a finite number of states, by definition. In contrast, a finite Petri net graph may result in a Petri net with an unbounded number of states. It should be noted that a finite-state automaton can always be represented as a Petri net; on the other hand, not all Petri nets can be represented as finite-state automata.

Similar to timed automata, we can define *timed Petri nets* by introducing a clock structure, except that now a clock sequence $\mathbf{v}_j$ is associated with a transition t_j. A positive real number, $v_{j,k}$, assigned to t_j has the following meaning: when transition t_j is enabled for the kth time, it does not fire immediately, but incurs a firing delay given by $v_{j,k}$; during this delay, tokens are kept in the input places of t_j. Not all transitions are required to have firing delays. Thus, we partition T into subsets T_0 and T_D, such that $T = T_0 \cup T_D$. T_0 is the set of transitions always incurring zero firing delay, and T_D is the set of transitions that generally incur some firing delay. The latter are called *timed transitions*. The *clock structure* (or *timing structure*) associated with a set of timed transitions $T_D \subseteq T$ of a marked Petri net (P, T, A, w, x) is a set $\mathbf{V} = \{\mathbf{v}_j : t_j \in T_D\}$ of clock (or lifetime) sequences $\mathbf{v}_j = \{v_{j,1}, v_{j,2}, \ldots\}$, $t_j \in T_D$, $v_{j,k} \in \mathbb{R}^+$, $k = 1, 2, \ldots$ A *timed Petri net* is a six-tuple $(P, T, A, w, x, \mathbf{V})$, where (P, T, A, w, x) is a marked Petri net and $\mathbf{V} = \{\mathbf{v}_j : t_j \in T_D\}$ is a clock structure. It is worth mentioning that this general structure allows for a variety of behaviors in a timed Petri net, including the possibility of multiple transitions being enabled at the same time or an enabled transition being preempted by the firing of another, depending on the values of the associated firing delays. The need to analyze and control such behavior in DES has motivated the development of a considerable body of analysis techniques for Petri net models which have been proven to be particularly suitable for this purpose (Moody and Antsaklis 1998; Peterson 1981).

Dioid Algebras

Another modeling framework is based on developing an algebra using two operations: $\min\{a, b\}$ (or $\max\{a, b\}$) for any real numbers a and b and addition $(a + b)$. The motivation comes from the observation that the operations "min" and "+" are the only ones required to develop the timed automaton model. Similarly, the operations "max" and "+" are the only ones used in developing the timed Petri net models described above. The operations are formally named *addition* and *multiplication* and denoted by $\oplus$ and $\otimes$

respectively. However, their actual meaning (in terms of regular algebra) is different. For any two real numbers a and b, we define

Addition : $\quad a \oplus b \equiv \max\{a, b\}$ (10)

Multiplication : $\quad a \otimes b \equiv a + b.$ (11)

This dioid algebra is also called a $(\max, +)$ algebra (Baccelli et al. 1992; Cuninghame-Green 1979). If we consider a standard linear discrete time system, its state equation is of the form

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k),$$

which involves (regular) multiplication $(\times)$ and addition $(+)$. It turns out that we can use a $(\max, +)$ algebra with DES, replacing the $(+, \times)$ algebra of conventional time-driven systems, and come up with a representation similar to the one above, thus paralleling to a considerable extent the analysis of classical time-driven linear systems. We should emphasize, however, that this particular representation is only possible for a subset of DES. Moreover, while conceptually this offers an attractive way to capture the event timing dynamics in a DES, from a computational standpoint, one still has to confront the complexity of performing the "max" operation when numerical information is ultimately needed to analyze the system or to design controllers for its proper operation.

Control and Optimization of Discrete Event Systems

The various control and optimization methodologies developed to date for DES depend on the modeling level appropriate for the problem of interest.

Logical Behavior. Issues such as ordering events according to some specification or ensuring the reachability of a particular state are normally addressed through the use of automata and Petri nets (Chen and Lafortune 1991; Moody and Antsaklis 1998; Ramadge and Wonham 1987). Supervisory control theory provides a systematic framework for formulating and solving problems of this type; a comprehensive coverage can be found in Cassandras and Lafortune (2008). Logical behavior issues are also encountered in the *diagnosis* of partially observed DES, a topic covered in the article ▶ Diagnosis of Discrete Event Systems.

Event Timing. When timing issues are introduced, timed automata and timed Petri nets are invoked for modeling purposes. Supervisory control in this case becomes significantly more complicated. An important class of problems, however, does not involve the ordering of individual events, but rather the requirement that selected events occur within a given "time window" or with some desired periodic characteristics. Models based on the algebraic structure of timed Petri nets or the $(\max, +)$ algebra provide convenient settings for formulating and solving such problems (Baccelli et al. 1992; Glasserman and Yao 1994).

Performance Analysis. As in classical control theory, one can define a performance (or cost) function as a means for quantifying system behavior. This approach is particularly crucial in the study of stochastic DES. Because of the complexity of DES dynamics, analytical expressions for such performance metrics in terms of controllable variables are seldom available. This has motivated the use of simulation and, more generally, the study of DES sample paths; these have proven to contain a surprising wealth of information for control purposes. The theory of *perturbation analysis* presented in the article ▶ Perturbation Analysis of Discrete Event Systems has provided a systematic way of estimating performance sensitivities with respect to system parameters (Cassandras and Lafortune 2008; Cassandras and Panayiotou 1999; Glasserman 1991; Ho and Cao 1991).

Discrete Event Simulation. Because of the aforementioned complexity of DES dynamics, simulation becomes an essential part of DES performance analysis (Law and Kelton 1991). Discrete event simulation can be defined as a

systematic way of generating sample paths of a DES by means of a timed automaton or its stochastic counterpart. The same process can be carried out using a Petri net model or one based on the dioid algebra setting.

Optimization. Optimization problems can be formulated in the context of both untimed and timed models of DES. Moreover, such problems can be formulated in both a deterministic and a stochastic setting. In the latter case, the ability to efficiently estimate performance sensitivities with respect to controllable system parameters provides a powerful tool for stochastic gradient-based optimization (when one can define derivatives) (Vázquez-Abad et al. 1998).

A treatment of all such problems from an application-oriented standpoint, along with further details on the use of the modeling frameworks discussed in this entry, can be found in the article ▶ Applications of Discrete-Event Systems.

Cross-References

- ▶ Applications of Discrete-Event Systems
- ▶ Diagnosis of Discrete Event Systems
- ▶ Discrete Event Systems and Hybrid Systems, Connections Between
- ▶ Modeling, Analysis, and Control with Petri Nets
- ▶ Perturbation Analysis of Discrete Event Systems
- ▶ Perturbation Analysis of Steady-State Performance and Sensitivity-Based Optimization
- ▶ Supervisory Control of Discrete-Event Systems

Bibliography

Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity. Wiley, Chichester/New York

Baeten JCM, Weijland WP (1990) Process algebra. Volume 18 of Cambridge tracts in theoretical computer science. Cambridge University Press, Cambridge/New York

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York

Cassandras CG, Panayiotou CG (1999) Concurrent sample path analysis of discrete event systems. J Discret Event Dyn Syst Theory Appl 9:171–195

Chen E, Lafortune S (1991) Dealing with blocking in supervisory control of discrete event systems. IEEE Trans Autom Control AC-36(6):724–735

Cuninghame-Green RA (1979) Minimax algebra. Number 166 in lecture notes in economics and mathematical systems. Springer, Berlin/New York

Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer Academic, Boston

Glasserman P, Yao DD (1994) Monotone structure in discrete-event systems. Wiley, New York

Ho YC (ed) (1991) Discrete event dynamic systems: analyzing complexity and performance in the modern world. IEEE, New York

Ho YC, Cao X (1991) Perturbation analysis of discrete event dynamic systems. Kluwer Academic, Dordrecht

Ho YC, Cassandras CG (1983) A new approach to the analysis of discrete event dynamic systems. Automatica 19:149–167

Hoare CAR (1985) Communicating sequential processes. Prentice-Hall, Englewood Cliffs

Hopcroft JE, Ullman J (1979) Introduction to automata theory, languages, and computation. Addison-Wesley, Reading

Law AM, Kelton WD (1991) Simulation modeling and analysis. McGraw-Hill, New York

Moody JO, Antsaklis P (1998) Supervisory control of discrete event systems using petri nets. Kluwer Academic, Boston

Peterson JL (1981) Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs

Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. SIAM J Control Optim 25(1):206–230

Vázquez-Abad FJ, Cassandras CG, Julka V (1998) Centralized and decentralized asynchronous optimization of stochastic discrete event systems. IEEE Trans Autom Control 43(5):631–655

M

Monotone Systems in Biology

David Angeli
Department of Electrical and Electronic Engineering, Imperial College London, London, UK
Dipartimento di Ingegneria dell'Informazione, University of Florence, Italy

Abstract

Mathematical models arising in biology might sometime exhibit the remarkable feature of preserving ordering of their solutions with

respect to initial data: in words, the "more" of x (the state variable) at time 0, the more of it at all subsequent times. Similar monotonicity properties are possibly exhibited also with respect to input levels. When this is the case, important features of the system's dynamics can be inferred on the basis of purely qualitative or relatively basic quantitative knowledge of the system's characteristics. We will discuss how monotonicity-related tools can be used to analyze and design biological systems with prescribed dynamical behaviors such as global stability, multistability, or periodic oscillations.

Keywords

Feedback interconnections; Monotone dynamics; Monotonicity checks

Introduction

Ordinary differential equations of a scalar unknown, under suitable assumptions for unicity of solutions, trivially enjoy the property that any pair of ordered initial conditions (according to the standard $\leq$ order defined for real numbers) gives rise to ordered solutions at all positive times (as well as negative, though this is less relevant for the developments that follow). Monotone systems are a special but significant class of dynamical models, possibly evolving in high-dimensional or even infinite-dimensional state spaces, that are nevertheless characterized by a similar property holding with respect to a suitably defined notion of partial order. They became the focus of considerable interest in mathematics after a series of seminal papers by Hirsch (1985, 1988) provided the basic definitions as well as deep results showing how generic convergence properties of their solutions are expected under suitable technical assumptions. Shortly before that Smale (1976), Smale's construction had already highlighted how specific solutions could instead exhibit

arbitrary behavior (including periodic or chaotic). Further results along these lines provide insight into which set of extra assumptions allow one to strengthen generic convergence to global convergence, including, for instance, existence of positive first integrals (Banaji and Angeli 2010; Mierczynski 1987), tridiagonal structure (Smillie 1984), or positive translation invariance (Angeli and Sontag 2008a).

While these tools were initially developed having in mind applications arising in ecology, epidemiology, chemistry, or economy, it was due to the increased importance of mathematical modeling in molecular biology and the subsequent rapid development of systems biology as an emerging independent field of investigation that they became particularly relevant to biology. The paper Angeli and Sontag (2003) first introduced the notion of control monotone systems, including input and output variables, that is of interest if one is looking at systems arising from interconnection of monotone modules. Small-gain theorems and related conditions were defined to study both positive (Angeli and Sontag 2004b) and negative (Angeli and Sontag 2003) feedback interconnections by relating their asymptotic behavior to properties of the discrete iterations of a suitable map, called the steady-state characteristic of the system.

In particular, convergence of this map is related to convergent solutions for the original continuous time system; on the other hand, specific negative feedback interconnections can instead give rise to oscillations as a result of Hopf bifurcations as in Angeli and Sontag (2008b) or to relaxation oscillators as highlighted in Gedeon and Sontag (2007).

A parallel line of investigation, originated in the work of Volpert et al. (1994), exploited the specific features of models arising in biochemistry by focusing on structural conditions for monotonicity of chemical reaction networks (Angeli et al. 2010; Banaji 2009). Monotonicity is only one of the possible tools adopted in the study of dynamics for such class of models in the related field of chemical reaction networks theory.

Mathematical Preliminaries

To illustrate the main tools of monotone dynamics, we consider the following systems defined on partially ordered input, state, and output spaces. Namely, along with the sets U, X, Y (which denote input, state, and output space, respectively), we consider corresponding partial orders $\succeq_X, \succeq_U, \succeq_Y$. A typical way of defining a partial order on a set S embedded in some Euclidean space E, $S \subset E$, is to first identify a cone K of positive vectors which belong to E. A cone in this context is any closed convex set which is preserved under multiplication times nonnegative scalars and such that $K \cap -K = \{0\}$. Accordingly we may denote $s_1 \succeq_S s_2$ whenever $s_1 - s_2 \in K$. A typical choice of K in the case of finite-dimensional $E = \mathbb{R}^n$ is the positive orthant, $(K = [0, +\infty)^n)$, in which case $\succeq$ can be interpreted as componentwise inequalities. More general orthants are also very useful in several applications as well as more exotic cones, smooth or polyhedral, according to the specific model considered. When dealing with input signals, we let $\mathcal{U}$ denote the set of locally essentially bounded and measurable functions of time. In particular, we inherit a partial order on $\mathcal{U}$ from the partial order on U according to the following definition:

$$u_1(\cdot) \succeq_\mathcal{U} u_2(\cdot) \ \Leftrightarrow \ u_1(t) \succeq_U u_2(t) \qquad \forall t \in \mathbb{R}.$$

When obvious from the context, we do not emphasize the space to which variables belong and simply write $\succeq$. Strict order notions are also of interest and especially relevant for some of the deepest implications of the theory. We let $s_1 \succ s_2$ denote $s_1 \succeq s_2$ and $s_1 \neq s_2$. While for partial orders induced by positivity cones, we let $s_1 \gg s_2$ denote $s_1 - s_2 \in \mathrm{int}(K)$.

A dynamical system is for us a continuous map $\varphi : \mathbb{R} \times X \to X$ which fulfills the property, $\varphi(0, x) = x$ for all $x \in X$ and $\varphi(t_2, \varphi(t_1, x)) = \varphi(t_1 + t_2, x)$ for all $t_1, t_2 \in \mathbb{R}$. Sometimes, when solutions are not globally defined (for instance, if the system is defined through a set of nonlinear differential equations), it is enough to restrict the definitions that follow to the domain of existence of solutions.

Definition 1 A monotone system φ is one that fulfills the following:

$$\forall x_1, x_2 \in X : x_1 \succeq x_2 \qquad \varphi(t, x_1) \succeq \varphi(t, x_2)$$
$$\forall t \geq 0. \qquad (1)$$

A system φ is strongly monotone when the following holds:

$$\forall x_1, x_2 \in X : x_1 \succ x_2 \qquad \varphi(t, x_1) \gg \varphi(t, x_2)$$
$$\forall t > 0. \qquad (2)$$

A control system is characterized by two continuous mappings: $\varphi : \mathbb{R} \times X \times \mathcal{U} \to X$ and the readout map $h : X \times U \to Y$.

Definition 2 A control system is monotone if

$$\forall u_1, u_2 \in \mathcal{U} : u_1 \succeq u_2, \quad \forall x_1, x_2 \in X : x_1 \succeq x_2,$$
$$\forall t \geq 0 \quad \varphi(t, x_1, u_1) \succeq \varphi(t, x_2, u_2) \quad (3)$$

and

$$\forall u_1, u_2 \in U : u_1 \succeq u_2, \quad \forall x_1, x_2 \in X : x_1 \succeq x_2,$$
$$h(x_1, u_1) \succeq h(x_2, u_2). \quad (4)$$

Notice that for any ordered state and input pairs x_1, x_2, u_1, u_2, the signals y_1 and y_2 defined as $y_1(t) := h(\varphi(t, x_1, u_1), u_1(t))$, $y_2(t) := h(\varphi(t, x_2, u_2), u_2(t))$ also fulfill, thanks to the Definition 2, $y_1(t) \succeq_Y y_2(t)$ (for all $t \geq 0$).

A system which is monotone with respect to the positive orthant is called cooperative. If a system is cooperative after reverting the direction of time, it is called competitive. Checking if a mathematical model specified by differential equations is monotone with respect to the partial order induced by some cone K is not too difficult. In particular, monotonicity, in its most basic formulation (1), simply amounts to a check of positive invariance of the set $\Gamma := \{(x_1, x_2) \in X^2 : x_1 \succeq x_2\}$ for a system formed by two copies of φ in parallel. This can be assessed without explicit knowledge of solutions, for instance, by using the notion of tangent cones and Nagumo's theorem (Angeli and Sontag 2003). Sufficient

conditions also exist to assess strong monotonicity, for instance, in the case of orthant cones. Finding whether there exists an order (as induced, for instance, by a suitable cone K) which can make a system monotone is instead a harder task which normally entails a good deal of insight in the systems dynamics.

It is worth mentioning that for the special case of linear systems, monotonicity is just equivalent to invariance of the cone K, as incremental properties (referred to pairs of solutions) are just equivalent to their non-incremental counterparts (referred to the 0 solution). In this respect, a substantial amount of theory exists starting from classical works such as the Perron-Frobenius theory on positive and cone-preserving maps; this is, however, outside the scope of this entry, and the interested reader may refer to Farina and Rinaldi (2000) for a recent book on the subject.

Monotone Dynamics

We divide this section in three parts; first we summarize the main tools for checking monotonicity with respect to orthant cones, then we recall some of the main consequences of monotonicity for the long-term behavior of solutions and, finally, we study interconnections of monotone systems.

Checking Monotonicity

Orthant cones and the partial orders they induce play a major role in biology applications. In fact, for systems described by equations

$$\dot{x} = f(x) \tag{5}$$

with $X \subset \mathbb{R}^n$ open and $f : X \to \mathbb{R}^n$ of class $\mathcal{C}^1$, the following characterization holds:

Proposition 1 *The system φ induced by the set of differential equations (5) is cooperative if and only if the Jacobian $\frac{\partial f}{\partial x}$ is a Metzler matrix for all $x \in X$.*

We recall that M is Metzler if $m_{ij} \geq 0$ for all $i \neq j$. Let $\Lambda = \text{diag}[\lambda_1, \ldots, \lambda_n]$ with $\lambda_i \in \{-1, 1\}$ and assume that the orthant $\mathcal{O} = \Lambda[0, +\infty)^n$. It is straightforward to see that $x_1 \succeq_{\mathcal{O}} x_2 \quad \Leftrightarrow$

$\Lambda x_1 \succeq \Lambda x_2$, where $\succeq$ denotes the partial order induced by the positive orthant, while $\succeq_{\mathcal{O}}$ denotes the order induced by $\mathcal{O}$. This means that we may check monotonicity with respect to $\mathcal{O}$ by performing a simple change of coordinates $z = \Lambda x$. As a corollary:

Proposition 2 *The system φ induced by the set of differential equations (5) is monotone with respect to $\succeq_{\mathcal{O}}$ if and only if $\Lambda \frac{\partial f}{\partial x} \Lambda$ is a Metzler matrix for all $x \in X$.*

Notice that conditions of Propositions 1 and 2 can be expressed in terms of sign constraints on off-diagonal entries of the Jacobian; in biological terms a sign constraint in an off-diagonal entry amounts to asking that a particular species (meaning chemical compound or otherwise) consistently exhibit throughout the considered model's state space either an excitatory or inhibitory effect on some other species of interest. Qualitative diagrams showing effects of species on each other are commonly used by biologists to understand the working principles of biomolecular networks.

Remarkably, Proposition 2 has also an interesting graph theoretical interpretation if one thinks of sign $\left(\frac{\partial f}{\partial x}\right)$ as the adjacency matrix of a graph with nodes $x_1 \ldots x_n$ corresponding to the state variables of the system.

Proposition 3 *The system φ induced by the set of differential equations (5) is monotone with respect to $\succeq_{\mathcal{O}}$ if and only if the directed graph of adjacency matrix sign $\left(\frac{\partial f}{\partial x}\right)$ (neglecting diagonal entries) has undirected loops with an even number of negative edges.*

This means in particular that $\frac{\partial f}{\partial x}$ must be sign symmetric (no predator-prey-type interactions) and in addition that a similar parity property has to hold on undirected loops of arbitrary length. Sufficient conditions for strong monotonicity are also known, for instance, in terms of irreducibility of the Jacobian matrix (Kamke's condition; see Hirsch and Smith 2003).

Asymptotic Dynamics
As previously mentioned, several important implications of monotonicity are with respect to

asymptotic dynamics. Let $\mathcal{E}$ denote the set of equilibria of φ. The following result is due to Hirsch (1985).

Theorem 1 *Let φ be a strongly monotone system with bounded solutions. There exists a zero measure set Q such that each solution starting in $X \setminus Q$ converges toward $\mathcal{E}$.*

Global convergence results can be achieved for important classes of monotone dynamics. For instance, when increasing conservation laws are present (see Banaji and Angeli 2010):

Theorem 2 *Let $X \subset K \subset \mathbb{R}^n$ be any two proper cones. Let φ on X be strongly monotone with respect to the partial order induced by K and preserving a K-increasing first integral. Then every bounded solution converges.*

Dually to first integrals, positive translation invariance of the dynamics also provide grounds for global convergence (see Angeli and Sontag 2008a):

Theorem 3 *If a system is strongly monotone and fulfills $\varphi(t, x_0 + hv) = \varphi(t, x_0) + hv$ for all $h \in \mathbb{R}$ and some $v \gg 0$, then all solutions with bounded projections in $v^{\perp}$ converge.*

The class of tridiagonal cooperative systems has also been investigated as a significant remarkable class of global convergent dynamics; see Smillie (1984). These arise from differential equations $\dot{x} = f(x)$ when $\partial f_i / \partial x_j = 0$ for all $|i - j| > 1$. Finally it is worth emphasizing how significant for biological systems, often subject to phenomena evolving at different timescales, are also results on singular perturbations (Gedeon and Sontag 2007; Wang and Sontag 2008).

Interconnected Monotone Systems
Results on interconnected monotone SISO systems are surveyed in Angeli and Sontag (2004a). The main tool used in this context is the notion of input-state and input–output steady-state characteristic.

Definition 3 A control system admits a well-defined input-state characteristic if for all constant inputs u there exists a unique globally asymptotically stable equilibrium $k_x(u)$ and the map $k_x(u)$ is continuous. If moreover the equilibrium is hyperbolic, then k_x is called a non-degenerate characteristic. The input–output characteristic is defined as $k_y(u) = h(k_x(u))$.

Let φ be system with a well-defined input–output characteristic k_y; we may define the iteration

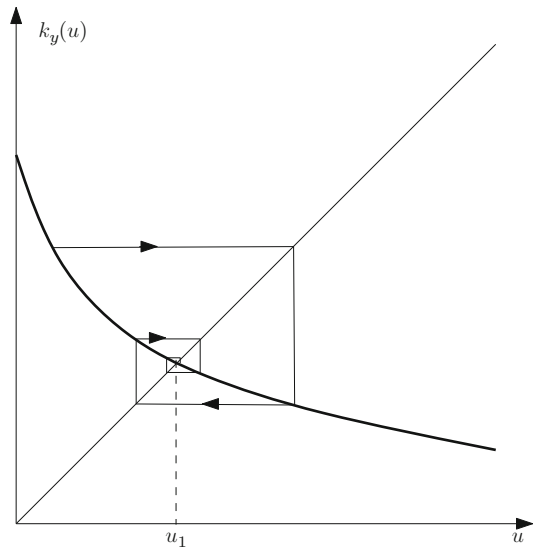$$u_{k+1} = k_y(u_k). \qquad (6)$$

It is clear that fixed points of (6) correspond to input values (and therefore to equilibria through the characteristic map k_x) of the closed-loop system derived by considering the unity feedback interconnection $u = y$. What is remarkable for monotone systems is that both in the case of positive and negative feedback and in a precise sense, stability properties of the fixed points of the discrete iteration (6) are matched by stability properties of the corresponding associated solutions of the original continuous time system. See Angeli and Sontag (2004b) for the case of positive feedback interconnections and Angeli et al. (2004) for applications of such results to synthesis and detection of multistability in molecular biology.

Multistability, in particular, is an important dynamical feature of specific cellular systems and can be achieved, with good degree of robustness with respect to different types of uncertainties, by means of positive feedback interconnections of monotone subsystems. The typical input–output characteristic k_y giving rise to such behavior is, in the SISO case, that of a sigmoidal function intersecting in 3 points the diagonal $u = y$. Two of the fixed points, namely, u_1 and u_3 (see Fig. 1), are asymptotically stable for (6), and the corresponding equilibria of the original continuous time monotone system are also asymptotically stable with a basin of attraction which covers almost all initial conditions. The fixed-point u_2 is unstable and the corresponding equilibrium is also such (under suitable technical assumption on the non-degeneracy of the I-O characteristic). Extensions of similar criteria to the MIMO case are presented in Enciso and Sontag (2005).
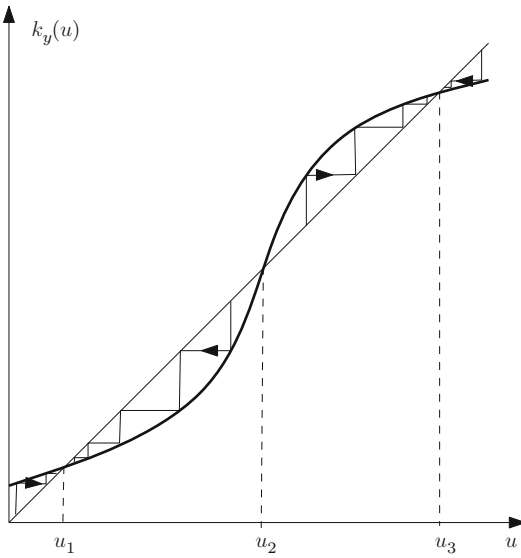
Monotone Systems in Biology, Fig. 1 Fixed points of a sigmoidal input–output characteristic

Monotone Systems in Biology, Fig. 2 Fixed point of a decreasing input–output characteristic

Negative feedback normally destroys monotonicity. As a result, the likelihood of complex dynamical behavior is highly increased. Nevertheless, input–output characteristics still can provide useful insight in the system's dynamics at least in the case of low feedback gain or, for high feedback gains, in the presence of sufficiently large delays. For instance, unity negative feedback interconnection of a SISO monotone system may give rise to a unique and globally asymptotically stable fixed point of (6), thanks to the decreasingness of the input–output characteristic and as shown in Fig. 2. Under such circumstances a small-gain result applies and global asymptotic stability of the corresponding equilibrium is guaranteed regardless of arbitrary input delays in the systems. See Angeli and Sontag (2003) for the simplest small-gain theorem developed in the context of SISO negative feedback interconnections of monotone systems and Enciso and Sontag (2006) for generalizations to systems with multiple inputs as well as delays. A generalization of small-gain results to the case of MIMO systems which are neither in a positive nor negative feedback configuration is presented in Angeli and Sontag (2011).

When the iteration (6) has an unstable fixed point, for instance, it converges to a period-2 solution, one may expect insurgence of oscillations around the equilibrium through a Hopf bifurcation provided sufficiently large delays in the input channels are allowed. This situation is analyzed in Angeli and Sontag (2008b) and illustrated through the study of the classical Golbeter's model for the *Drosophila*'s circadian rhythm.

Summary and Future Directions

Verifying that a control system preserves some ordering of initial conditions provides important and far-reaching implications for its dynamics. Insurgence of specific behaviors can often be inferred on the basis of purely qualitative knowledge (as in the case of Hirsch's generic convergence theorem) as well as additional basic quantitative knowledge as in the case of positive and negative feedback interconnections of monotone systems. For the above reasons, applications in molecular biology of monotone system's theory are gradually emerging: for instance, in the study of MAPK cascades or circadian oscilla-

tions, as well as in Chemical Reaction Networks Theory. Generally speaking, while monotonicity as a whole cannot be expected in large networks, experimental data shows that the number of negative feedback loops in biological regulatory networks is significantly lower than in a random signed graph of comparable size Maayan et al. (2008).

Analyzing the properties of monotone dynamics may potentially lead to better understanding of the key regulatory mechanisms of complex networks as well as the development of bottom-up approaches for the identification of meaningful submodules in biological networks. Potential research directions may include both novel computational tools and specific applications to systems biology, for instance:

- Algorithms for detection of monotonicity with respect to exotic orders (such as arbitrary polytopic cones or even state-dependent cones)
- Application of monotonicity-based ideas to control synthesis (see, for instance, Aswani and Tomlin (2009) where the special class of piecewise affine systems is considered)

Cross-References

▶ Deterministic Description of Biochemical Networks
▶ Spatial Description of Biochemical Networks
▶ Stochastic Description of Biochemical Networks

Recommended Reading

For readers interested in the mathematical details of monotone systems theory we recommend the following:

  Smith H (1995) Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems. Mathematical surveys and monographs, vol 41. AMS, Providence

A more recent technical survey of aspects related to asymptotic dynamics of monotone systems is

Hirsch MW, Smith H (2005) Monotone dynamical systems (Chapter 4). In: Canada A, Drábek P, Fonda A (eds) Handbook of differential equations ordinary differential equations, vol 2. Elsevier

Bibliography

Angeli D, Sontag ED (2003) Monotone control systems. IEEE Trans Autom Control 48:1684–1698

Angeli D, Sontag ED (2004a) Interconnections of monotone systems with steady-state characteristics. In: Optimal control, stabilization and nonsmooth analysis. Lecture notes in control and information sciences, vol 301. Springer, Berlin, pp 135–154

Angeli D, Sontag ED (2004b) Multi-stability in monotone input/output systems. Syst Control Lett 51:185–202

Angeli D, Sontag ED (2008a) Translation-invariant monotone systems, and a global convergence result for enzymatic futile cycles. Nonlinear Anal Ser B Real World Appl 9:128–140

Angeli D, Sontag ED (2008b) Oscillations in I/O monotone systems. IEEE Trans Circuits Syst 55:166–176

Angeli D, Sontag ED (2011) A small-gain result for orthant-monotone systems in feedback: the non sign-definite case. Paper appeared in the 50th IEEE conference on decision and control, Orlando, 12–15 Dec 2011

Angeli D, Ferrell JE, Sontag ED (2004) Detection of multistability, bifurcations, and hysteresis in a large class of biological positive-feedback systems. Proc Natl Acad Sci USA 101:1822–1827

Angeli D, de Leenheer P, Sontag ED (2010) Graph-theoretic characterizations of monotonicity of chemical networks in reaction coordinates. J Math Biol 61:581–616

Aswani A, Tomlin C (2009) Monotone piecewise affine systems. IEEE Trans Autom Control 54:1913–1918

Banaji M (2009) Monotonicity in chemical reaction systems. Dyn Syst 24:1–30

Banaji M, Angeli D (2010) Convergence in strongly monotone systems with an increasing first integral. SIAM J Math Anal 42:334–353

Banaji M, Angeli D (2012) Addendum to "Convergence in strongly monotone systems with an increasing first integral". SIAM J Math Anal 44:536–537

Enciso GA, Sontag ED (2005) Monotone systems under positive feedback: multistability and a reduction theorem. Syst Control Lett 54:159–168

Enciso GA, Sontag ED (2006) Global attractivity, I/O monotone small-gain theorems, and biological delay systems. Discret Contin Dyn Syst 14:549–578

Enciso GA, Sontag ED (2008) Monotone bifurcation graphs. J Biol Dyn 2:121–139

Farina L, Rinaldi S (2000) Positive linear systems: theory and applications. Wiley, New York

M

Gedeon T, Sontag ED (2007) Oscillations in multi-stable monotone systems with slowly varying feedback. J Differ Equ 239:273–295

Hirsch MW (1985) Systems of differential equations that are competitive or cooperative II: convergence almost everywhere. SIAM J Math Anal 16:423–439

Hirsch MW (1988) Stability and convergence in strongly monotone dynamical systems. Reine Angew Math 383:1–53

Hirsch MW, Smith HL (2003) Competitive and cooperative systems: a mini-review. In: Positive systems, Rome. Lecture notes in control and information science, vol 294. Springer, pp 183–190

Maayan A, Iyengar R, Sontag ED (2008) Intracellular regulatory networks are close to monotone systems. IET Syst Biol 2:103–112

Mierczynski J (1987) Strictly cooperative systems with a first integral. SIAM J Math Anal 18:642–646

Smale S (1976) On the differential equations of species in competition. J Math Biol 3:5–7

Smillie J (1984) Competitive and cooperative tridiagonal systems of differential equations. SIAM J Math Anal 15:530–534

Volpert AI, Volpert VA, Volpert VA (1994) Traveling wave solutions of parabolic Systems. Translations of mathematical monographs, vol 140. AMS, Providence

Wang L, Sontag ED (2008) Singularly perturbed monotone systems and an application to double phosphorylation cycles. J Nonlinear Sci 18:527–550

Motion Description Languages and Symbolic Control

Sean B. Andersson
Mechanical Engineering and Division of
Systems Engineering, Boston University,
Boston, MA, USA

Abstract

The fundamental idea behind symbolic control is to mitigate the complexity of a dynamic system by limiting the set of available controls to a typically finite collection of symbols. Each symbol represents a control law that may be either open or closed loop. With these symbols, a simpler description of the motion of the system can be created, thereby easing the challenges of analysis and control design. In this entry, we provide a high-level description of symbolic control; discuss briefly its history, connections, and applications; and provide a few insights into where the field is going.

Keywords

Abstraction; Complex systems; Formal methods

Introduction

Systems and control theory is powerful paradigm for analyzing, understanding, and controlling dynamic systems. Traditional tools in the field for developing and analyzing control laws, however, face significant challenges when one needs to deal with the complexity that arises in many practical, real-world settings such as the control of autonomous, mobile systems operating in uncertain and changing physical environments. This is particularly true when the tasks to be achieved are not easily framed in terms of motion to a point in the state space. One of the primary goals of symbolic control is to mitigate this complexity by abstracting some combination of the system dynamics, the space of control inputs, and the physical environment to a simpler, typically finite, model.

This fundamental idea, namely, that of abstracting away the complexity of the underlying dynamics and environment, is in fact a quite natural one. Consider, for example, how you give instructions to another person wanting to go to a point of interest. It would be absurd to provide details at the level of their actuators, namely, with commands to their individual muscles (or to carry the example to an even more absurd extreme, to the dynamic components that make up those muscles). Rather, very high-level commands are given, such as "follow the road," "turn right," and so on. Each of these provides a description of what to do with the understanding that the person can carry out those commands in their own fashion. Similarly, the environment itself is abstracted, and only elements meaningful to the task at hand are described. Thus, continuing the example above, rather than

providing metric information or a detailed map, the instructions may use environmental features to determine when particular actions should be terminated and the next begun, such as "follow the road until the second intersection, then turn right."

Underlying the idea of symbolic control is the notion that rich behaviors can result from simple actions. This premise was used in many early robots and can be traced back at least to the ideas of Norbert Wiener on cybernetics (see Arkin 1998). It is at the heart of the behavior-based approach to robotics (Brooks 1986). Similar ideas can also be seen in the development of a high-level language (G-codes) for Computer Numerically Controlled (CNC) machines. The key technical ideas in the more general setting of symbolic control for dynamic systems can be traced back to Brockett (1988) which introduced ideas of formalizing a modular approach to programming motion control devices through the development of a Motion Description Language (MDL).

The goal of the present work is to introduce the interested reader to the general ideas of symbolic control as well as to some of its application areas and research directions. While it is not a survey paper, a few select references are provided throughout to point the reader in hopefully fruitful directions into the literature.

Models and Approaches

There are at least two related but distinct approaches to symbolic control. Both begin with a mathematical description of the system, typically given as an ordinary differential equation of the form

$$\dot{x} = f(x, u, t), \qquad y = h(x, t) \qquad (1)$$

where x is a vector describing the state of the system, y is the output of the sensors of the system, and u is the control input.

Under the first approach to symbolic control, the focus is on reducing the complexity of the space of possible control signals by limiting the system to a typically finite collection of control symbols. Each of these symbols represents a control law that may be open loop or may utilize output feedback. For example, *follow the road* could be a feedback control law that uses sensor measurements to determine the position relative to the road and then applies steering commands so that the system stays on the road while simultaneously maintaining a constant speed. There are, of course, many ways to accomplish the specifics of this task, and the details will depend on the particular system. Thus, an autonomous four-wheeled vehicle equipped with a laser range finder, an autonomous motorcycle equipped with ultrasonic sensors, or an autonomous aerial vehicle with a camera would each carry out the command in their own way, and each would have very different trajectories. They would all, however, satisfy the notion of *follow the road*. Description of the behavior of the system can then be given in terms of the abstract symbols rather than in terms of the details of the trajectories.

Typically each of these symbols describes an action that at least conceptually is simple. In order to generate rich motions to carry out complex tasks, the system is switched between the available symbols. Switching conditions are often referred to as *interrupts*. Interrupts may be purely time-based (e.g., apply a given symbol for T seconds) or may be expressed in terms of symbols representing certain environmental conditions. These may be simple function of the measurements (e.g., interrupt when an intersection is detected) or may represent more complicated scenarios with history and dynamics (e.g., interrupt after the second intersection is detected). Just as the input symbols abstract away the details of the control space and of the motion of the system, the interrupt symbols abstract away the details of the environment. For example, *intersection* has a clear high-level meaning but a very different sensor "signature" for particular systems.

As a simple illustrative example, consider a collection of control symbols designed for moving along a system of roads, {*follow road*, *turn right*, *turn left*}, and a collection of interrupt
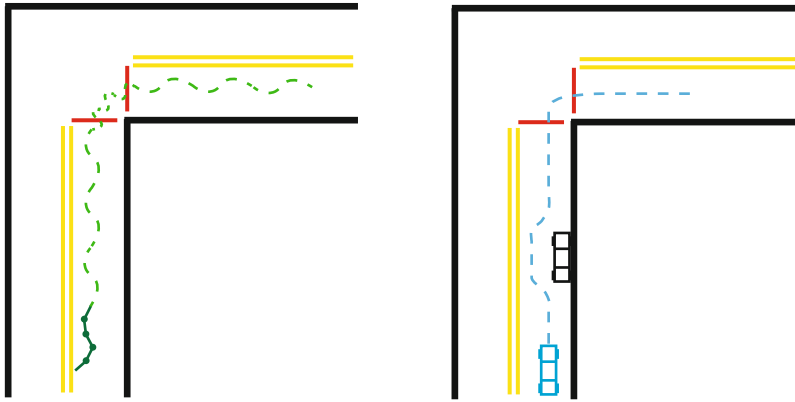
Motion Description Languages and Symbolic Control, Fig. 1 Simple example of symbolic control with a focus on abstracting the inputs. Two systems, a snakelike robot and an autonomous car, are given a high-level plan in terms of symbols for navigating a right-hand turn. The systems each interpret the same symbols in their own ways, leading to different trajectories due both to differences in dynamics and also to different sensors cues as caused, for example, by the parked vehicle encountered by the car in this scenario

symbols for triggering changes in such a setting, {*in intersection*, *clear of intersection*}. Suppose there are two vehicles that can each interpret these symbols, an autonomous car and a snake-like robot, as illustrated in Fig. 1. It is reasonable to assume that the control symbols each describe relatively complex dynamics that allow, for example, for obstacle avoidance while carrying out the action. Figure 1 illustrates a possible situation where the two systems carry out the plan defined by the symbolic sequence:

(*Follow the road* UNTIL *in intersection*)
(*Turn right* UNTIL *clear of intersection*)

The intent of this plan is for the system to navigate a right-hand turn. As shown in the figure, the actual trajectories followed by the systems can be markedly different due in part to system dynamics (the snakelike robot undulates, while the car does not) as well as to different sensor responses (when the car goes through, there is a parked vehicle that it must navigate around, while the snakelike robot found a clear path during its execution). Despite these differences, both systems achieve the goal of the plan.

The collection of control and interrupt symbols can be thought of as a language for describing and specifying motion and are used to write programs that can be compiled into an executable for a specific system. Different rules for doing this can be established that define different languages, analogous to different high-level programming languages such as C++, Java, or Python. Further details can be found in, for example, Manikonda et al. (1998).

Under the second approach, the focus is on representing the dynamics and state space (or environment) of the system in an abstract, symbolic way. The fundamental idea is to lump all the states in a region into a single abstract element and to then represent the entire system with a finite number of these elements. Control laws are then defined that steer all the states in one element into some state in a different region. The symbolic control system is then the finite set of elements representing regions together with the finite set of controllers for moving between them. It can be thought of essentially as a graph (or more accurately as a transition system) in which the nodes represent regions in the state space and the edges represent achievable transitions between them. The goal of this abstraction step is for the two representations to be equivalent (or at least approximately equivalent) in that any motion that can be achieved in one can be achieved in the other (in an appropriate sense). Planning and analysis can then

Abstraction   Symbolic model and planning   Execution of plan
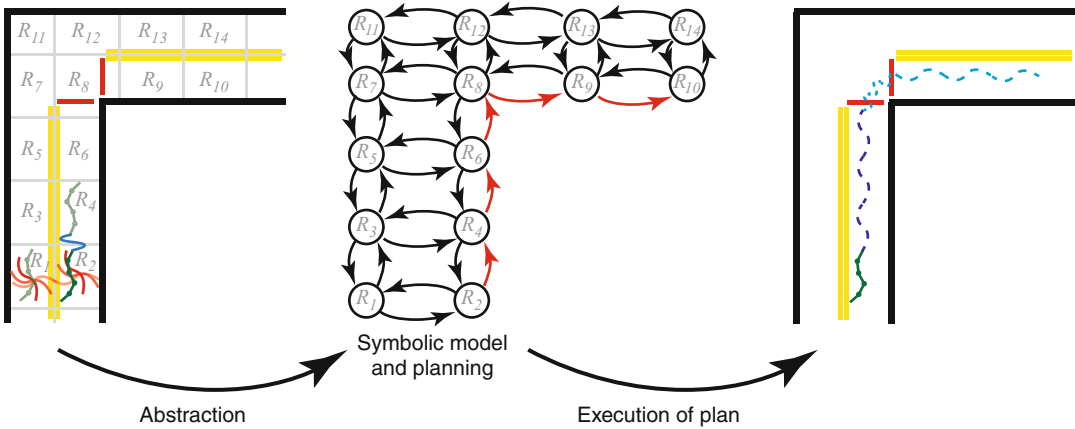
Motion Description Languages and Symbolic Control, Fig. 2 Simple example of symbolic control with a focus on abstracting the system dynamics and environment. The initial environment (*left image*) is segmented into different regions and simple controllers developed for moving from region to region. The image shows two possible controllers: one that actuates the robot through a tight slither pattern to move forward by one region and one that twists the robot to face the cell to the left before slithering across and then reorienting. The combination of regions and actions yields a symbolic abstraction (*center image*) that allows for planning to achieve specific goals, such as moving through the right-hand turn. Executing this plan leads to a physical trajectory of the system (*right image*)

be done on the (simpler) symbolic model. Further details on such schemes can be found in, for example, Tabuada (2006) and Bicchi et al. (2006).

As an illustrative example, consider as before a snakelike robot moving through a right-hand turn. In a simplified view of this second approach to symbolic control, one begins by dividing the environment up into regions and then defining controllers to steer the robot from region to region as illustrated in the left image in Fig. 2. This yields the symbolic model shown in the center of Fig. 2. A plan is then developed on this model to move from the initial position to the final position. This planning step can take into account restrictions on the motion and subgoals of the task. Here, for example, one may want the robot to stay to the right of the double yellow line that is in its lane of traffic. The plan $R_2 \rightarrow R_4 \rightarrow R_6 \rightarrow R_8 \rightarrow R_9 \rightarrow R_{10}$ is one sequence that drives the system around the turn while satisfying the lane requirement. Each transition in the sequence corresponds to a control law. The plan is then executed by applying the sequence of control laws, resulting in the actual trajectory shown in the right image in Fig. 2.

Applications and Connections

The fundamental idea behind symbolic control, namely, mitigating complexity by abstracting a system, its environment, and even the tasks to be accomplished into a simpler but (approximately) equivalent model, is a natural and a powerful one. It has clear connections to both hybrid systems (Brockett 1993; Egerstedt 2002) and to quantized control (Bicchi et al. 2006), and the tools from those fields are often useful in describing and analyzing systems with symbolic representations of the control and of the dynamics. Symbolic control is not, however, strictly a subcategory of either field, and it provides a unique set of tools for the control and analysis of dynamic systems.

Brockett's original MDL was intended to serve as a tool for describing and planning robot motion. Inspired in part by this, languages for motion continue to be developed. Some of these extend and provide a more formal basis for motion programming (Manikonda et al. 1998) and interconnection of dynamic systems into a single whole (Murray et al. 1992), while some are designed for specialized dynamics or applications such as flight vehicles

M

(Frazzoli et al. 2005), self-assembly (Klavins 2007), and other areas. In addition to studying standard systems and control theoretic ideas, including notions of reachability (Bicchi et al. 2002) and stability (Tarraf et al. 2008), the framework of symbolic control introduces interesting questions such as how to understand the reduction of complexity that can be achieved for a given collection of symbols (Egerstedt and Brockett 2003).

While there are many application areas of symbolic control, the one that is perhaps most active is that of motion planning for autonomous mobile robots (Belta et al. 2007). As illustrated in Figs. 1 and 2, symbolic control allows the planning problem (i.e., the determination of how to achieve a desired task) to be separated from the complexities of the dynamics. The approach has been particularly fertile when coupled with symbolic descriptions of the tasks to be achieved. While point-to-point commands are useful, and can be often thought of as symbols themselves from which to build more complicated commands, most tasks that one would want mobile robots to carry out involve combinations of spatial goals (move to a certain location), sequencing (first do this and then do that) or other temporal requirements (repeatedly visit a collection of regions), as well as safety or other restrictions (avoid obstacles or regions that are dangerous for the robot to traverse). Such tasks can be described using a variety of temporal logics. These are, essentially, logic systems that include rules related to time in addition to the standard Boolean operators. These tasks can be combined with a symbolic description of a system and then automated tools used both to check whether the system is able to perform the desired task and to design plans that ensure the system will do so (Fainekos et al. 2009). To ensure that results on the abstract, symbolic system are valid on the original dynamic system, methods exist for guaranteeing the equivalence of the two models, in an appropriate sense (Girard and Pappas 2007).

Summary and Future Directions

Symbolic control proceeds from the basic goal of mitigating the complexity of dynamic systems, especially in real-world scenarios, to yield a simplification of the problems of analysis and control design. It builds upon results from diverse fields while also contributing new ideas to those areas, including hybrid system theory, formal languages and grammars, and motion planning. There are many open, interesting questions that are the subject of ongoing investigations as well as the genesis of future research.

One particularly fruitful direction is that of combining symbolic control with stochasticity. Systems that operate in the real world are subject to noise with respect both to their inputs (noisy actuators) and to their outputs (noisy sensors). Recent work along these lines can be found in the formal methods approach to motion planning and in hybrid systems (Abate et al. 2011; Lahijanian et al. 2012). The fundamental idea is to use a Markov chain, Markov decision process, or similar model as the symbolic abstraction and then, as in all symbolic control, to do the analysis and planning on this simpler model.

Another interesting direction is to address questions of optimality with respect to the symbols and abstractions for a given dynamic system. Of course, the notion of "optimal" must be made clear, and there are several reasonable notions one could define. There is a clear trade-off between the complexity of individual symbols, the number of symbols used in the motion "alphabet," and the complexity in terms of, say, average number of symbols required to code programs that achieve a given set of tasks. The complexity of a necessary alphabet is also related to the variety of tasks the system might need to perform. An autonomous vacuuming robot is likely to need far fewer symbols in its library than an autonomous vehicle that must operate in everyday traffic conditions and respond to unusual events such as traffic jams. The question of the "right" set of symbols can also

be of use in efficient descriptions of motion in domains such as dance (Baillieul and Ozcimder 2012).

It is intuitively clear that to handle complex scenarios and environments, a hierarchical approach is likely needed. Organizing symbols into progressively higher levels of abstraction should allow for more efficient reasoning, planning, and reaction to real-world settings. Such structures already appear in existing works, such as in the behavior-based approach of Brooks (1986), in the extended Motion Description Language in Manikonda et al. (1998), and in the Spatial Semantic Hierarchy of Kuipers (2000). Despite these efforts, there is still a need for a rigorous approach for analyzing and designing symbolic hierarchical systems.

The final direction discussed here is that of the connection of symbolic control to emergent behavior in large groups of dynamic agents. There are a variety of intriguing examples in nature in which large numbers of agents following simple rules produce large-scale, coherent behavior, including in fish schools and termite and ant colonies (Johnson 2002). How can one predict the global behavior that will emerge from a large collection of independent agents following simple rules (symbols)? How can one design a set of symbols to produce a desired collective behavior? While there has been some work in symbolic control for self-assembling systems (Klavins 2007), this general topic remains a rich area for research.

Cross-References

Recommended Reading

Brockett's original paper Brockett (1988) is a surprisingly short but informational read. More thorough descriptions can be found in Manikonda et al. (1998) and Egerstedt (2002). An excellent description of symbolic control in robotics, particularly in the context of temporal logics and formal methods, can be found in Belta et al. (2007). There are also several related articles in a 2011 special issue of the IEEE Robotics and Automation magazine (Kress-Gazit 2011).

Bibliography

Abate A, D'Innocenzo A, Di Benedetto MD (2011) Approximate abstractions of stochastic hybrid systems. IEEE Trans Autom Control 56(11):2688–2694

Arkin RC (1998) Behavior-based robotics. MIT, Cambridge

Baillieul J, Ozcimder K (2012) The control theory of motion-based communication: problems in teaching robots to dance. In: American control conference, Montreal, pp 4319–4326

Belta C, Bicchi A, Egerstedt M, Frazzoli E, Klavins E, Pappas GJ (2007) Symbolic planning and control of robot motion [Grand Challenges of Robotics]. IEEE Robot Autom Mag 14(1):61–70

Bicchi A, Marigo A, Piccoli B (2002) On the reachability of quantized control systems. IEEE Trans Autom Control 47(4):546–563

Bicchi A, Marigo A, Piccoli B (2006) Feedback encoding for efficient symbolic control of dynamical systems. IEEE Trans Autom Control 51(6):987–1002

Brockett RW (1988) On the computer control of movement. In: IEEE International conference on robotics and automation, Philadelphia, pp 534–540

Brockett RW (1993) Hybrid models for motion control systems. In: Trentelman HL, Willems JC (eds) Essays on control. Birkhauser, Boston, pp 29–53

Brooks R (1986) A robust layered control system for a mobile robot. IEEE J Robot Autom RA-2(1):14–23

Egerstedt M (2002) Motion description languages for multi-modal control in robotics. In: Bicchi A, Cristensen H, Prattichizzo D (eds) Control problems in robotics. Springer, pp 75–89

Egerstedt M, Brockett RW (2003) Feedback can reduce the specification complexity of motor programs. IEEE Trans Autom Control 48(2):213–223

Fainekos GE, Girard A, Kress-Gazit H, Pappas GJ (2009) Temporal logic motion planning for dynamic robots. Automatica 45(2):343–352

Frazzoli E, Dahleh MA, Feron E (2005) Maneuver-based motion planning for nonlinear systems with symmetries. IEEE Trans Robot 21(6):1077–1091

M

Girard A, Pappas GJ (2007) Approximation metrics for discrete and continuous systems. IEEE Trans Autom Control 52(5):782–798

Johnson S (2002) Emergence: the connected lives of ants, brains, cities, and software. Scribner, New York

Klavins E (2007) Programmable self-assembly. IEEE Control Syst 27(4):43–56

Kress-Gazit H (2011) Robot challenges: toward development of verification and synthesis techniques [from the Guest Editors]. IEEE Robot Autom Mag 18(3):22–23

Kuipers B (2000) The spatial semantic hierarchy. Artif Intell 119(1–2):191–233

Lahijanian M, Andersson SB, Belta C (2012) Temporal logic motion planning and control with probabilistic satisfaction guarantees. IEEE Trans Robot 28(2):396–409

Manikonda V, Krishnaprasad PS, Hendler J (1998) Languages, behaviors, hybrid architectures, and motion control. In: Baillieul J, Willems JC (eds) Mathematical control theory. Springer, New York, pp 199–226

Murray RM, Deno DC, Pister KSJ, Sastry SS (1992) Control primitives for robot systems. IEEE Trans Syst Man Cybern 22(1):183–193

Tabuada P (2006) Symbolic control of linear systems based on symbolic subsystems. IEEE Trans Autom Control 51(6):1003–1013

Tarraf DC, Megretski A, Dahleh MA (2008) A framework for robust stability of systems over finite alphabets. IEEE Trans Autom Control 53(5):1133–1146

Motion Planning for Marine Control Systems

Andrea Caiti

DII – Department of Information Engineering & Centro "E. Piaggio", ISME – Interuniversity Research Centre on Integrated Systems for the Marine Environment, University of Pisa, Pisa, Italy

Abstract

In this chapter we review motion planning algorithms for ships, rigs, and autonomous marine vehicles. Motion planning includes path and trajectory generation, and it goes from optimized route planning (off-line long-range path generation through operating research methods) to reactive on-line trajectory reference generation, as given by the guidance system. Crucial to the marine systems case is the presence of environmental external forces (sea state, currents, winds) that drive the optimized motion generation process.

Keywords

Configuration space; Dynamic programming; Grid search; Guidance controller; Guidance system; Maneuvering; Motion plan; Optimization algorithms; Path generation; Route planning; Trajectory generation; World space

Introduction

Marine control systems include primarily ships and rigs moving on the sea surface, but also underwater systems, manned (submarines) or unmanned, and eventually can be extended to any kind of off-shore moving platform.

A *motion plan* consists in determining what motions are appropriate for the marine system to reach a goal, or a target/final state (LaValle 2006). Most often, the final state corresponds to a geographical location or destination, to be reached by the system while respecting constraints of physical and/or economical nature. Motion planning in marine systems hence starts from *route planning*, and then it covers desired *path generation* and *trajectory generation*. Path generation involves the determination of an ordered sequence of states that the system has to follow; trajectory generation requires that the states in a path are reached at a prescribed time.

Route, path, and trajectory can be generated off-line or on-line, exploiting the feedback from the system navigation and/or from external sources (weather forecast, etc.). In the feedback case, planning overlaps with the *guidance system*, i.e., the continuous computation of the reference (desired) state to be used as reference input by the motion control system (Fossen 2011).

Formal Definitions and Settings

Definitions and classifications as in Goerzen et al. (2010) and Petres et al. (2007) are followed

throughout the section. The marine systems under considerations live in a physical space referred to as the *world space* (e.g., a submarine lives in a 3-D Euclidean space). A *configuration* q is a vector of variables that define position and orientation of the system in the world space. The set of all possible configurations is the *configuration space*, or *C-space*. The vector of configuration and configuration rate of changes is the *state* of the system $\mathbf{x} = \begin{bmatrix} q^T \dot{q}^T \end{bmatrix}^T$, and the set of all the possible states is the *state space*. The kino-dynamic model associated to the system is represented by the system *state equations*. The regions of *C-space* free from obstacles are called *C-free*.

The *path planning* problem consists in determining a curve $\gamma : [0, 1] \rightarrow C\text{-free}, s \rightarrow \gamma(s)$, with $\gamma(0)$ corresponding to the initial configuration and $\gamma(1)$ corresponding to the goal configuration. Both initial and goal configurations are in *C-free*. The *trajectory planning* problem consists in determining a curve γ and a *time law*: $t \rightarrow s(t)$ *s.t.* $\gamma(s) = \gamma(s(t))$. In both cases, either the path or the trajectory must be compatible with the system state equations. In the following, definitions of motion algorithm properties are given referring to path planning, but the same definitions can be easily extended to trajectory planning.

A motion planning algorithm is *complete* if it finds a path when one exists, and returns a proper flag when no path exists. The algorithm is *optimal* when it provides the path that minimizes some cost function J. The (strictly positive) cost function J is isotropic, when it depends only on the system configuration ($J = J(q)$), or anisotropic, when it depends also on an external force field $\mathbf{f}$ (e.g., sea currents, sea state, weather perturbations) ($J = J(q, \mathbf{f})$). The cost function J induces a *pseudometric* in the configuration space; the distance d between configurations q_1 and q_2 through the path γ is the "cost-to-go" from q_1 to q_2 along γ:

$$d(q_1, q_2) = \int_0^1 J(\gamma_{q_1 q_2}(s), \mathbf{f}) ds \qquad (1)$$

An optimal motion planning problem is:

– *Static* if there is perfect knowledge of the environment at any time, *dynamic* otherwise
– *Time-invariant* when the environment does not evolve (e.g., coastline that limits the *C-free* subspace), *time-variant* otherwise (e.g., other systems – ships, rigs – in navigation)
– *Differentially constrained* if the system state equations act as a constraint on the path, *differentially unconstrained* otherwise

In practice, optimal motion planning problems are solved numerically through discretization of the *C-space*. *Resolution* completeness/optimality of an algorithm implies the achievement of the solution as the discretization interval tends to zero. *Probabilistic* completeness/optimality implies that the probability of finding the solution tends to 1 as the computation time approaches infinity. *Complexity* of the algorithm refers to the computational time required to find a solution as a function of the dimension of the problem.

The scale of the motion w.r. to the scale of the system defines the specific setting of the problem. In cargo ships route planning from one port call to the next, the problem is stated first as static, time-invariant, differentially unconstrained *path* planning problem; once a large-scale route is thus determined, it can be refined on smaller scales, e.g., smoothing it, to make it compatible with ship maneuverability. Maneuvering the same cargo ship in the approaches to a harbor has to be casted as a dynamic, time-variant, differentially constrained *trajectory* planning problem.

Large-Scale, Long-Range Path Planning

Route determination is a typical long-range path planning problem for a marine system. The geographical map is discretized into a grid, and the optimal path between the approaches of the starting and destination ports is determined as a sequence of adjacent grid nodes. The problem is taken as time-invariant and differentially unconstrained, at least in the first stages of the procedure. It is assumed that the ship will cruise at its own (constant) most economical speed to optimize bunker consumption, the major

M

source of operating costs (Wang and Meng 2012). Navigation constraints (e.g., allowed ship traffic corridors for the given ship class) are considered, as well as weather forecasts and predicted/prevailing currents and winds. The cost-to-go Eq. (1) is built between adjacent nodes either in terms of time to travel or in terms of operating costs, both computed correcting the nominal speed with the environmental forces. Optimality is defined in terms of shortest time/minimum operating cost; the anisotropy introduced by sea/weather conditions is the driving element of the optimization, making the difference with respect to straightforward shortest route computation. The approach is iterated, starting with a coarse grid and then increasing grid resolution in the neighborhood of the previously found path.

The most widely used optimization approach for surface ships is *dynamic programming* (LaValle 2006); alternatively, since the determination of the optimal path along the grid nodes is equivalent to a search over a graph, the *A* algorithm* is applied (Delling et al. 2009). As the discretization grid gets finer, system dynamics are introduced, accounting for ship maneuverability and allowing for deviation from the constant ship speed assumption. Dynamic programming allows to include system dynamics at any level of resolution desired; however, when system dynamics are considered, the problem dimension grows from 2-D to 3-D (2-D space plus time).

In the case of underwater navigation, path planning takes place in a 3-D world space, and the inclusion of system dynamics makes it a 4-D problem; moreover, bathymetry has to be included as an additional constraint to shape the *C-free* subspace. Dynamic programming may become unfeasible, due to the increase in dimensionality. Computationally feasible algorithms for this case include global search strategies with probabilistic optimality, as *genetic algorithms* (Alvarez et al. 2004), or improved grid-search methods with resolution optimality, as FM* (Petres et al. 2007).

Environmental force fields are intrinsically dynamic fields; moreover, the prediction of such fields at the moment of route planning may be updated as the ship is in transit along the route. The path planning algorithms can/must be rerun, over a grid in the neighborhood of the nominal path, each time new environmental information becomes available. Kino-dynamic model of the ship must be included, allowing for deviation from the established path and ship speed variation around the nominal most economical speed. The latter case is particularly important: increasing/decreasing the speed to avoid a weather perturbation keeping the same route may indeed result in a reduced operating cost with respect to path modifications keeping a constant speed. This implies that the timing over the path must be specified. Dynamic programming is well suited for this transition from path to trajectory generation, and it is still the most commonly used approach to trajectory (re)planning in reaction to environmental predictions update.

When discretizing the world space, the minimum grid size should still be large enough to allow for ship maneuvering between grid nodes. This is required for safety, to allow evasive maneuvering when other ships are at close ranges, and for the generation of smooth, dynamics-compliant trajectories between grid points. This latter aspect bridges motion planning with guidance.

Trajectory Planning, Maneuvering Generation, and Guidance Systems

Once a path has been established over a spatial grid, a continuous reference has to be generated, linking the nodes over the grid. The generation of the reference trajectory has to take into account all the relevant dynamic properties and constraints of the marine system, so that the reference motion is *feasible*. In this scenario, the path/trajectory nodes are *way-points*, and the trajectory generation connects the way-points along the route. The approaches to trajectory generation can be divided between those that do not compute explicitly in advance the whole trajectory and those that do.

Among the approaches that do not need explicit trajectory computation between way-points,
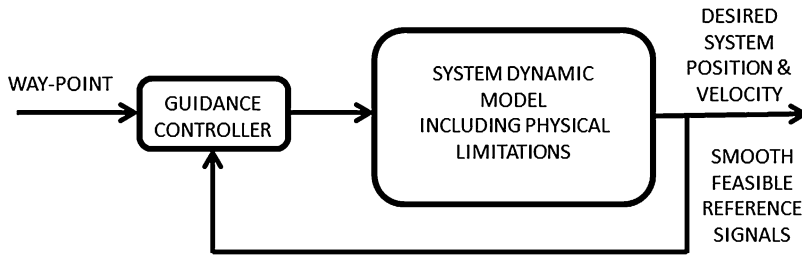
Motion Planning for Marine Control Systems, Fig. 1 Generation of a reference trajectory with a system model and a guidance controller (Adapted from Fossen (2011))

the most common is the *line-of-sight* (LOS) guidance law (Pettersen and Lefeber 2001). LOS guidance can be considered a path generation, more than a trajectory generation, since it does not impose a time law over the path; it computes directly the desired ship reference heading on the basis of the current ship position and the previous and next way-point positions. A review of other guidance approaches can be found in Breivik and Fossen (2008), where maneuvering along the path and steering around the way-points are also discussed. From such a set of different maneuvers, a library of motion primitives can be built (Greytak and Hover 2010), so that any motion can be specified as a sequence of primitives. While each primitive is feasible by construction, an arbitrary sequence of primitives may not be feasible. An optimized search algorithm (dynamic programming, A*) is again needed to determine the optimal feasible maneuvering sequence.

Path/trajectory planning explicitly computing the desired motion among two way-points may include a system dynamic model, or may not. In the latter case, a sufficiently smooth curve that connects two way-points is generated, for instance, as splines or as Dubins paths (LaValle 2006). Curve generation parameters must be set so that the "sufficiently smooth" part is guaranteed. After curve generation, a trim velocity is imposed over the path (path planning), or a time law is imposed, e.g., smoothly varying the system reference velocity with the local curvature radius.

Planners that do use a system dynamic model are described in Fossen (2011) as part of the guidance system. In practice, the dynamic model

is used in simulation, with a (simulated) feedback controller (*guidance controller*), the next way-point as input, and the (simulated) system position and velocity as output. The simulated results are feasible maneuvers by construction and can be given as reference position/velocity to the physical control system (Fig. 1).

Summary and Future Directions

Motion planning for marine control systems employs methodological tools that range from operating research to guidance, navigation, and control systems. A crucial role in marine applications is played by the anisotropy induced by the dynamically changing environmental conditions (weather, sea state, winds, currents – the external force fields). The quality of the plan will depend on the quality of environmental information and predictions.

While motion planning can be considered a mature issue for ships, rigs, and even standalone autonomous vehicles, current and future research directions will likely focus on the following items:

– Coordinated motion planning and obstacle avoidance for *teams* of autonomous surface and underwater vehicles (Aguiar and Pascoal 2012; Casalino et al. 2009)
– Naval traffic regulation compliant maneuvering in restricted spaces and collision evasion maneuvering (Tam and Bucknall 2010)
– Underwater intervention robotics (Antonelli 2006; Sanz et al. 2010)

M

Cross-References

▶ Control of Networks of Underwater Vehicles
▶ Mathematical Models of Marine Vehicle-Manipulator Systems
▶ Mathematical Models of Ships and Underwater Vehicles
▶ Underactuated Marine Control Systems

Recommended Reading

Motion planning is extensively treated in LaValle (2006), while the essential reference on marine control systems is the book by Fossen (2011). Goerzen et al. (2010) reviews motion planning algorithms in terms of computational properties. The book Antonelli (2006) includes the treatment of planning and control in intervention robots. The papers Breivik and Fossen (2008) and Tam et al. (2009) provide a survey of both terminology and guidance design for both open and close space maneuvering. In particular, Tam et al. (2009) links motion planning to navigation rules.

Bibliography

Aguiar AP, Pascoal A (2012) Cooperative control of multiple autonomous marine vehicles: theoretical foundations and practical issues. In: Roberts GN, Sutton R (eds) Further advances in unmanned marine vehicles. IET, London, pp 255–282

Alvarez A, Caiti A, Onken R (2004) Evolutionary path planning for autonomous underwater vehicles in a variable ocean. IEEE J Ocean Eng 29(2):418–429

Antonelli G (2006) Underwater robots – motion and force control of vehicle-manipulator systems. 2nd edn. Springer, Berlin/New York

Breivik M, Fossen TI (2008) Guidance laws for planar motion control. In: Proceedings of the 47th IEEE conference on decision & control, Cancun

Casalino G, Simetti E, Turetta A (2009) A three-layered architecture for real time path planning and obstacle avoidance for surveillance usvs operating in harbour fields. In: Proceedings of the IEEE oceans'09-Europe, Bremen

Delling D, Sanders P, Schultes D, Wagner D (2009) Engineering route planning algorithms. In Lerner J, Wagner D, Zweig KA (eds) Algorithmics of large and complex networks. Lecture notes in computer science, vol 5515. Springer, Berlin/New York, pp 117–139

Fossen TI (2011) Handbook of marine crafts hydrodynamics and motion control. Wiley, Chichester

Goerzen C, Kong Z, Mettler B (2010) A survey of motion planning algorithms from the perspective of autonomous UAV guidance. J Intell Robot Syst 57:65–100

Greytak MB, Hover FS (2010) Robust motion planning for marine vehicle navigation. In: Proceedings of the 18th international offshore & polar engineering conference, Vancouver

LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge/New York. Available on-line at http://planning.cs.uiuc.edu. Accessed 6 June 2013

Petres C, Pailhas Y, Patron P, Petillot Y, Evans J, Lane D (2007) Path planning for autonomous underwater vehicles. IEEE Trans Robot 23(2):331–341

Pettersen KY, Lefeber E (2001) Way-point tracking control of ships. In: Proceedings of the 40th IEEE conference decision & control, Orlando

Sanz PJ, Ridao P, Oliver G, Melchiorri C, Casalino G, Silvestre C, Petillot Y, Turetta A (2010) TRIDENT: a framework for autonomous underwater intervention missions with dexterous manipulation capabilities. In: Proceedings of the 7th IFAC symposium on intelligent autonomous vehicles, Lecce

Tam CK, Bucknall R (2010) Path-planning algorithm for ships in close-range encounters. J Mar Sci Technol 15:395–407

Tam CK, Bucknall R, Greig A (2009) Review of collision avoidance and path planning methods for ships in close range encounters. J Navig 62:455–476

Wang S, Meng Q (2012) Liner ship route schedule design with sea contingency time and port time uncertainty. Transp Res B 46:615–633

Motion Planning for PDEs

Thomas Meurer
Faculty of Engineering,
Christian-Albrechts-University Kiel, Kiel,
Germany

Abstract

Motion planning refers to the design of an open-loop or feedforward control to realize prescribed desired paths for the system states or outputs. For distributed-parameter systems described by partial differential equations (PDEs), this requires to take into account the spatial-temporal system dynamics. Here, flatness-based techniques provide

a systematic inversion-based motion planning approach, which is based on the parametrization of any system variable by means of a flat or basic output. With this, the motion planning problem can be solved rather intuitively as is illustrated for linear and semilinear PDEs.

Keywords

Basic output; Flatness; Formal integration; Formal power series; Trajectory assignment; Trajectory planning; Transition path

Introduction

Motion planning or trajectory planning refers to the design of an open-loop control to realize prescribed desired temporal or spatial-temporal paths for the system states or outputs. Examples include smart structures with embedded actuators and sensors such as adaptive optics in telescopes, adaptive wings or smart skins, thermal and reheating processes in steel industry, and deep drawing, start-up, shutdown, or transitions between operating points in chemical engineering, as well as multi-agent deployment and formation control (see, e.g., the overview in Meurer 2013).

For the solution of the motion planning and tracking control problem for finite-dimensional linear and nonlinear systems, differential flatness as introduced in Fliess et al. (1995) has evolved into a well-established inversion-based technique. Differential flatness implies that any system variable can be parametrized in terms of a flat or a so-called basic output and its time derivatives up to a problem-dependent order. As a result, the assignment of a suitable desired trajectory for the flat output directly yields the respective state and input trajectories to realize the prescribed motion. Flatness can be adapted to systems governed by partial differential equations (PDEs). For this, different techniques have been developed utilizing operational calculus or spectral theory for linear PDEs, (formal) power series for linear PDEs, and PDEs involving polynomial

nonlinearities as well as formal integration for semilinear PDEs using a generalized Cauchy-Kowalevski approach. To illustrate the principle ideas and the evolving research results starting with Fliess et al. (1997), subsequently different techniques are introduced based on selected example problems. For this, the exposition is primarily restricted to parabolic PDEs with a brief discussion of motion planning for hyperbolic PDEs before concluding with possible future research directions.

Linear PDEs

In the following, a scalar linear diffusion-reaction equation is considered in the state variable $x(z, t)$ with boundary control $u(t)$ governed by

$$\partial_t x(z, t) = \partial_z^2 x(z, t) + r x(z, t) \qquad \text{(1a)}$$

$$\partial_z x(0, t) = 0, \quad x(1, t) = u(t) \qquad \text{(1b)}$$

$$x(z, 0) = 0. \qquad \text{(1c)}$$

This PDE describes a wide variety of thermal and fluid systems including heat conduction and tubular reactors. Herein, $r \in \mathbb{R}$ refer to the reaction coefficient and the initial state is without loss of generality assumed zero. In order to solve the motion planning problem for (1), a feedforward control $t \mapsto u^*(t)$ is determined to realize a finite-time transition between the initial state and a final stationary state $x_T^*(z)$ to be imposed for $t \geq T$.

Formal Power Series
By making use of the formal power series expansion of the state variable

$$x(z, t) \to \hat{x}(z, t) = \sum_{n=0}^{\infty} \hat{x}_n(t) \frac{z^n}{n!} \qquad \text{(2)}$$

the evaluation of (1) results in the 2nd-order recursion

$$\hat{x}_n(t) = \partial_t \hat{x}_{n-2}(t) - r x_{n-2}(t), \quad n \geq 2 \quad \text{(3a)}$$

$$\hat{x}_1(t) = 0. \qquad \text{(3b)}$$

In order to be able to solve (3) for $\hat{x}_n(t)$, it is hence required to impose $\hat{x}_0(t) = \hat{x}(0,t)$. Denoting $y(t) = x(0,t)$ or respectively

$$\hat{x}_0(t) = y(t) \tag{3c}$$

implies

$$\hat{x}_{2n}(t) = (\partial_t - r)^n \circ y(t), \quad \hat{x}_{2n+1}(t) = 0. \tag{4}$$

Hence, any series coefficient in (2) can be differentially parametrized by means of $y(t)$. Taking into account the inhomogeneous boundary condition in (1b), i.e.,

$$u(t) = x(1,t) = \sum_{n=0}^{\infty} \frac{x_n(t)}{n!} = \sum_{n=0}^{\infty} \frac{x_{2n}(t)}{(2n)!} \tag{5}$$

yields that $y(t) = x(0,t)$ can be considered as a flat or basic output. In particular, by prescribing a suitable trajectory $t \mapsto y^*(t) \in C^\infty(\mathbb{R})$ for $y(t)$, the evaluation of (5) yields the feedforward control $u^*(t)$ which is required to realize the spatial-temporal path $x^*(z,t)$ obtained from the substitution of $y^*(t)$ into (2) with coefficients parametrized by (4). This, however, relies on the uniform convergence of (2) in view of (4) with at least a unit radius of convergence in z. For this, the notion of a Gevrey class function is needed (Rodino 1993).

Definition 1 (Gevrey class) The function $y(t)$ is in $G_{D,\alpha}(\Lambda)$, the Gevrey class of order α in $\Lambda \subseteq \mathbb{R}$, if $y(t) \in C^\infty(\Lambda)$ and for every closed subset Λ' of Λ there exists a $D > 0$ such that $\sup_{t \in \Lambda'} |\partial_t^n y(t)| \le D^{n+1}(n!)^\alpha$.

The set $G_{D,\alpha}(\Lambda)$ forms a linear vector space and a ring with respect to the arithmetic product of functions which is closed under the standard rules of differentiation. Gevrey class functions of order $\alpha < 1$ are entire and are analytic if $\alpha = 1$.

Theorem 1 Let $y(t) \in G_{D,\alpha}(\mathbb{R})$ for $\alpha < 2$, then the formal power series (2) with coefficients (4) converges uniformly with infinite radius of convergence.

The proof of this result can be, e.g., found in Laroche et al. (2000) and Lynch and Rudolph (2002) and relies on the analysis of the recursion (3) taking into account the assumptions on the function $y(t)$.

Trajectory Assignment

To apply these results for the solution of the motion planning problem to achieve finite-time transitions between stationary profiles, it is crucial to properly assign the desired trajectory $y^*(t)$ for the basic output $y(t)$. For this, observe that stationary profiles $x^s(z) = x^s(z; y^s)$ are due to the flatness property (Classically stationary solutions are to be defined in terms of stationary input values $x^s(1) = u^s$.) governed by

$$0 = \partial_z^2 x^s(z) + r x^s(z) \tag{6a}$$

$$\partial_z x^s(0) = 0, \quad x^s(0) = y^s. \tag{6b}$$

Hence, assigning different y^s results in different stationary profiles $x^s(z; y^s)$. The connection between an initial stationary profile $x_0^s(z; y_0^s)$ and a final stationary profile $x_T^s(z; y_T^s)$ is achieved by assigning $y^*(t)$ such that

$$y^*(0) = y_0^s, \qquad y^*(T) = y_T^s$$
$$\partial_t^n y^*(0) = 0, \qquad \partial_t^n y^*(T) = 0, \quad n \ge 1.$$

This implies that $y^*(t)$ has to be locally nonanalytic at $t \in \{0, T\}$ and in view of the previous discussion has thus to be a Gevrey class function of order $\alpha \in (1, 2)$. For specific problems different functions have been suggested fulfilling these properties. In the following, the ansatz

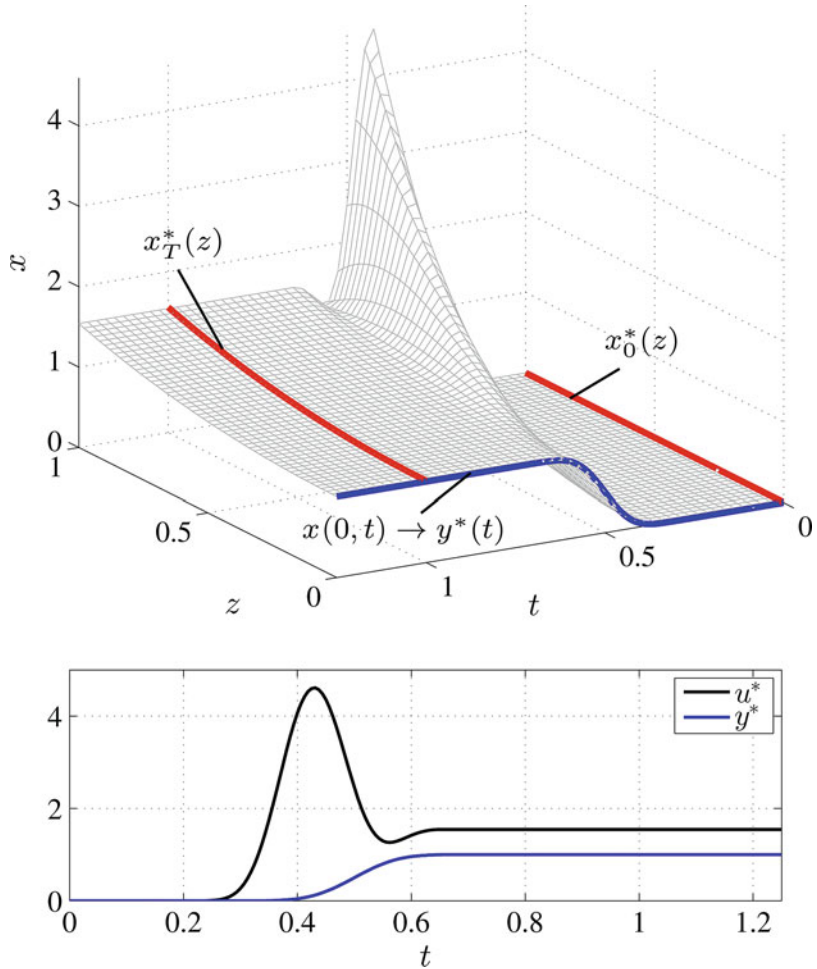$$y^*(t) = y_0^s + (y_T^s - y_0^s)\Phi_{T,\gamma}(t) \tag{7a}$$

is used with

$$\Phi_{T,\gamma}(t) = \begin{cases} 0, & t \le 0 \\ \frac{\int_0^t h_{T,\gamma}(\tau)\mathrm{d}\tau}{\int_0^T h_{T,\gamma}(\tau)\mathrm{d}\tau} & t \in (0, T) \\ 1, & t \ge T \end{cases} \tag{7b}$$

for $h_{T,\gamma}(t) = \exp\left(-[t/T(1 - t/T)]^{-\gamma}\right)$ if $t \in (0, T)$ and $h_{T,\gamma}(t) = 0$ else. It can be shown that

Motion Planning for PDEs, Fig. 1 Simulated spatial-temporal transition path (*top*) and applied flatness-based feedforward control $u^*(t)$ and desired trajectory $y^*(t)$ (*bottom*) for (1)

(7b) is a Gevrey class function of order $\alpha = 1 + 1/\gamma$ (Fliess et al. 1997). Alternative functions are presented, e.g., in Rudolph (2003).

Simulation Example

In order to illustrate the results of the motion planning procedure described above, let $r = -1$ in (1). The differential parametrization (4) of the series coefficients is evaluated for the desired trajectory $y^*(t)$ defined in (7) for $y_0^s = 0$ and $y_T^s = 1$ with the transition time $T = 1$ and the parameter $\gamma = 2$. With this, the finite-time transition between the zero initial stationary profile $x_0^*(z) = 0$ and the final stationary profile $x_T^*(z) = x_T^s(z) = y_T^s \cosh(z)$ is realized along the trajectory $x(0, t) = y^*(t)$. The corresponding

feedforward control and spatial-temporal transition path are shown in Fig. 1.

Extensions and Generalizations

The previous considerations constitute a first systematic approach to solve motion planning problems of systems governed by PDEs. The underlying techniques can be, however, further generalized to address coupled systems of PDEs, certain classes of nonlinear PDEs (see also section "Semilinear PDEs"), or in-domain control.

While the application of formal power series is restricted to boundary control diffusion-convection-reaction systems, the approach can be combined with so-called resummation techniques to overcome convergence issues such as slowly

converging or even divergent series expansions (Laroche et al. 2000; Meurer and Zeitz 2005).

Flatness-based techniques for motion planning can be also embedded into an operator theoretic context using semigroup theory by restricting the analysis to so-called Riesz spectral operators. This enables to analyze coupled systems of linear PDEs with both boundary and in-domain control in a single and multiple spatial coordinates with a common framework (Meurer 2011, 2013). In addition, experimental results for flexible beam and plate structures with embedded piezoelectric actuators confirm the applicability of this design approach and the achievable high tracking accuracy when transiently shaping the deflection profile (Schröck et al. 2013).

Semilinear PDEs

Flatness can be extended to semilinear PDEs. This is subsequently illustrated for the diffusion-reaction system

$$\partial_t x(z, t) = \partial_z^2 x(z, t) + r(x(z, t)) \qquad (8a)$$

$$\partial_z x(0, t) = 0, \quad x(1, t) = u(t) \qquad (8b)$$

$$x(z, 0) = 0 \qquad (8c)$$

with boundary input $u(t)$. Similar to the previous section, the motion planning problem refers to the determination of a feedforward control $t \mapsto u^*(t)$ to realize finite-time transitions starting at the initial profile $x_0^*(z) = x(z, 0) = 0$ to achieve a final stationary profile $x_T^*(z)$ for $t \geq T$.

Formal Power Series
If $r(x(z, t))$ is a polynomial in $x(z, t)$ or an analytic function, then similar to the previous section, formal power series can be applied to solve the motion planning problem. This, however, relies on the successive evaluation of Cauchy's product formula. As an example, consider

$$r(x(z, t)) = r_1 x(z, t) + r_2 x^2(z, t),$$

then the formal power series ansatz (2) results in the recursion

$$\hat{x}_n(t) = \partial_t \hat{x}_{n-2}(t) - r_1 x_{n-2}(t)$$
$$- r_2 \sum_{j=0}^{n-2} \binom{n}{j} \hat{x}_j(t) \hat{x}_{n-j}(t), \quad n \geq 2 \qquad (9a)$$

$$\hat{x}_1(t) = 0. \qquad (9b)$$

Similar to the linear setting in the section "Linear PDEs" above, the recursion can be solved for $\hat{x}_n(t)$ by imposing $\hat{x}_0(t) = \hat{x}(0, t)$ or respectively

$$\hat{x}_0(t) = y(t). \qquad (9c)$$

As a result, also in this nonlinear setting any series coefficient can be expressed in terms of $y(t)$ and its time derivatives. Hence, $y(t) = x(0, t)$ denotes a basic output for the semilinear PDE (8). The uniform series convergence can be analyzed by restricting any trajectory $y(t)$ to a certain Gevrey order α while simultaneously restricting the absolute values of d, r_1 and r_2 (Dunbar et al. 2003; Lynch and Rudolph 2002). These restrictions can be approached using, e.g., resummation techniques to sum slowly converging or divergent series to a meaningful limit. The reader is therefore referred to Meurer and Zeitz (2005) or Meurer and Krstic (2011), with the latter introducing a PDE-based approach for formation control of multi-agent systems.

Formal Integration
A generalization of these results has been recently suggested in Schörkhuber et al. (2013) by making use of an abstract Cauchy-Kowalevski theorem in Gevrey classes. In order to illustrate this, solve (8a) for $\partial_z^2 x(z, t)$ and formally integrate with respect to z taking into account the boundary conditions (8b). This yields the implicit solution

$$x(z, t) = x_+(0, t) \int_0^z \int_0^p \big[\partial_t x(q, t)$$
$$- r(x(q, t)) \big] \mathrm{d}q \, \mathrm{d}p \qquad (10a)$$

$$u(t) = x(1, t), \qquad (10b)$$

which can be used to develop to a flatness-based design systematics for motion planning given semilinear PDEs. For this, introduce

$$y(t) = x(0, t), \tag{11}$$

and rewrite (10b) in terms of the sequence of functions $(x^{(n)}(z, t))_{n=0}^{\infty}$ according to

$$x^{(0)}(z, t) = y(t) \tag{12a}$$

$$x^{(n+1)}(z, t) = x^{(0)}(z, t) + \int_0^z \int_0^p \big[\partial_t x^{(n)}(q, t)$$

$$-r(x^{(n)}(q, t))\big] \mathrm{d}q \mathrm{d}p. \tag{12b}$$

From this, it is obvious that $y(t)$ denotes a basic output differentially parametrizing the state variable $x(z, t) = \lim_{n \to \infty} x^{(n)}(z, t)$ and the boundary input $u(t) = x(1, t)$ provided that the limit exists as $n \to \infty$. As is shown in Schörkhuber et al. (2013) by making use of scales of Banach spaces in Gevrey classes and abstract Cauchy-Kowalevski theory, the convergence of the parametrized sequence of functions $(x^{(n)}(z, t))_{n=0}^{\infty}$ can be ensured in some compact subset of the domain $z \in [0, 1]$. Besides its general setup this approach provides an iteration scheme, which can be directly utilized for a numerically efficient solution of the motion planning problem.

Simulation Example

Let the reaction be subsequently described by

$$r(x(z, t)) = \sin(2\pi x(z, t)). \tag{13}$$

The iterative scheme (12) is evaluated for the desired trajectory $y^*(t)$ defined in (7) for $y_0^s = 0$ and $y_T^s = 1$ with the transition time $T = 1$ and the parameter $\gamma = 1$, i.e., the desired trajectory is of Gevrey order $\alpha = 2$. The resulting feedforward control $u^*(t)$ and the spatial-temporal transition path resulting from the numerical solution of the PDE are depicted in Fig. 2. The desired finite-time transition between the zero initial stationary profile $x_0^*(z) = 0$ and the final stationary profile $x_T^*(z) = x_T^s(z)$ determined by

$$0 = \partial_z^2 x^s(z) + r(x^s(z)) \tag{14a}$$

$$\partial_z x^s(0) = 0, \quad x^s(0) = y^s. \tag{14b}$$

is clearly achieved along the prescribed path $y^*(t)$.

Extensions and Generalizations

Generalizations of the introduced formal integration approach to solve motion planning problems for systems of coupled PDEs are, e.g., provided in Schörkhuber et al. (2013). Moreover, linear diffusion-convection-reaction systems with spatially and time-varying coefficients defined on a higher-dimensional parallelepipedon are addressed in Meurer and Kugi (2009) and Meurer (2013).

Hyperbolic PDEs

Hyperbolic PDEs exhibiting wavelike dynamics require the development of a design systematics explicitly taking into account the finite speed of wave propagation. For linear hyperbolic PDEs, operational calculus has been successfully applied to determine the state and input parametrizations in terms of the basic output and its advanced and delayed arguments (Petit and Rouchon 2001, 2002; Rouchon 2001; Rudolph and Woittennek 2008; Woittennek and Rudolph 2003). In addition, the method of characteristics can be utilized to address both linear and quasi-linear hyperbolic PDEs. Herein, a suitable change of coordinates enables to reformulate the PDE in a normal form, which can be (formally) integrated in terms of a basic output. With this, also an efficient numerical procedure can be developed to solve motion planning problems for hyperbolic PDEs (Woittennek and Mounier 2010).

Summary and Future Directions

Motion planning constitutes an important design step when solving control problems for systems governed by PDEs. This is particularly due to

Motion Planning for PDEs, Fig. 2 Simulated spatial-temporal transition path (*top*) and applied flatness-based feedforward control $u^*(t)$ and desired trajectory $y^*(t)$ (*bottom*) for (8) with (13)
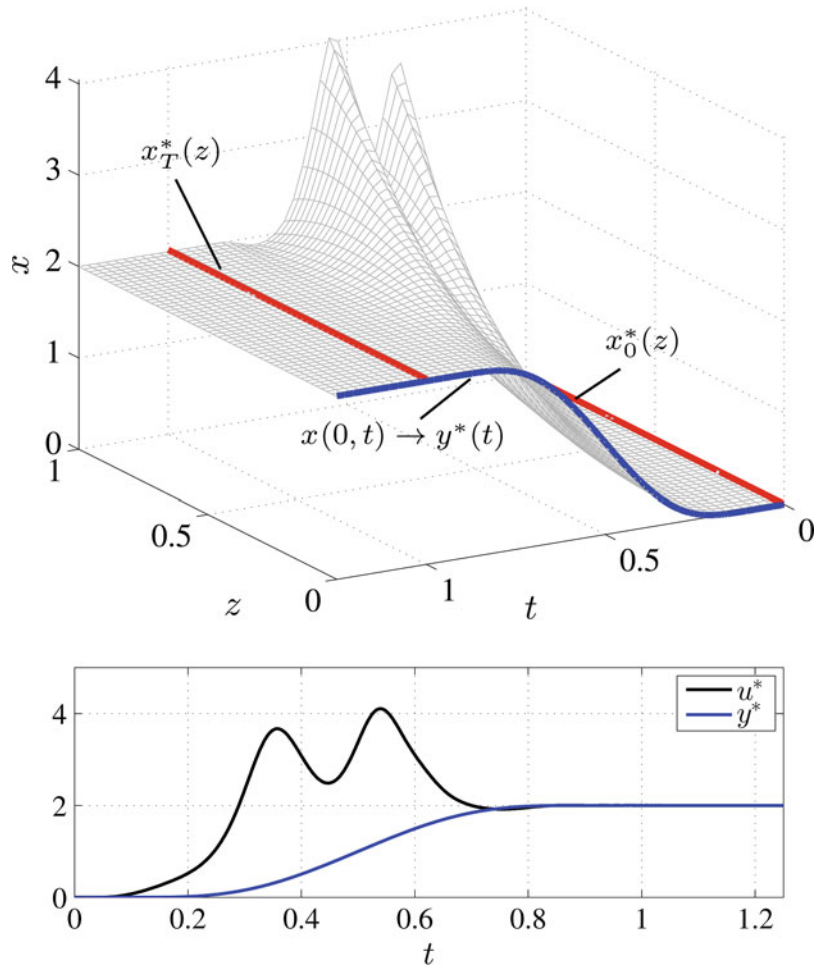
the increasing demands on quality, accuracy, and efficiency, which require to turn away from the pure stabilization of an operating point toward the realization of specific start-up, transition, or tracking tasks. In view of these aspects, future research directions might deepen and further evolve the following:

– Semi-analytic design techniques taking into account suitable approximation schemes for complex-shaped spatial domains
– Nonlinear PDEs and coupled systems of nonlinear PDEs with boundary and in-domain control
– Applications arising, e.g., in aeroelasticity, micromechanical systems, fluid flow, and fluid-structure interaction.

Cross-References

▶ Boundary Control of 1-D Hyperbolic Systems
▶ Boundary Control of Korteweg-de Vries and Kuramoto–Sivashinsky PDEs
▶ Control of Fluids and Fluid-Structure Interactions

Bibliography

Dunbar W, Petit N, Rouchon P, Martin P (2003) Motion planning for a nonlinear Stefan problem. ESAIM Control Optim Calculus Var 9:275–296
Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of non–linear systems: introductory theory and examples. Int J Control 61:1327–1361

Fliess M, Mounier H, Rouchon P, Rudolph J (1997) Systèmes linéaires sur les opérateurs de Mikusiński et commande d'une poutre flexible. ESAIM Proc 2:183–193

Laroche B, Martin P, Rouchon P (2000) Motion planning for the heat equation. Int J Robust Nonlinear Control 10:629–643

Lynch A, Rudolph J (2002) Flatness-based boundary control of a class of quasilinear parabolic distributed parameter systems. Int J Control 75(15):1219–1230

Meurer T (2011) Flatness-based trajectory planning for diffusion-reaction systems in a parallelepipedon – a spectral approach. Automatica 47(5):935–949

Meurer T (2013) Control of higher-dimensional PDEs: flatness and backstepping designs. Communications and control engineering series. Springer, Berlin

Meurer T, Krstic M (2011) Finite-time multi-agent deployment: a nonlinear PDE motion planning approach. Automatica 47(11):2534–2542

Meurer T, Kugi A (2009) Trajectory planning for boundary controlled parabolic PDEs with varying parameters on higher-dimensional spatial domains. IEEE Trans Autom Control 54(8): 1854–1868

Meurer T, Zeitz M (2005) Feedforward and feedback tracking control of nonlinear diffusion-convection-reaction systems using summability methods. Ind Eng Chem Res 44:2532–2548

Petit N, Rouchon P (2001) Flatness of heavy chain systems. SIAM J Control Optim 40(2):475–495

Petit N, Rouchon P (2002) Dynamics and solutions to some control problems for water-tank systems. IEEE Trans Autom Control 47(4):594–609

Rodino L (1993) Linear partial differential operators in gevrey spaces. World Scientific, Singapore

Rouchon P (2001) Motion planning, equivalence, and infinite dimensional systems. Int J Appl Math Comput Sci 11:165–188

Rudolph J (2003) Flatness based control of distributed parameter systems. Berichte aus der Steuerungs– und Regelungstechnik. Shaker–Verlag, Aachen

Rudolph J, Woittennek F (2008) Motion planning and open loop control design for linear distributed parameter systems with lumped controls. Int J Control 81(3):457–474

Schörkhuber B, Meurer T, Jüngel A (2013) Flatness of semilinear parabolic PDEs – a generalized Cauchy-Kowalevski approach. IEEE Trans Autom Control 58(9):2277–2291

Schröck J, Meurer T, Kugi A (2013) Motion planning for Piezo–actuated flexible structures: modeling, design, and experiment. IEEE Trans Control Syst Technol 21(3):807–819

Woittennek F, Mounier H (2010) Controllability of networks of spatially one-dimensional second order P.D.E. – an algebraic approach. SIAM J Control Optim 48(6):3882–3902

Woittennek F, Rudolph J (2003) Motion planning for a class of boundary controlled linear hyperbolic PDE's involving finite distributed delays. ESAIM Control Optim Calculus Var 9: 419–435

Motorcycle Dynamics and Control

Martin Corless
School of Aeronautics & Astronautics, Purdue University, West Lafayette, IN, USA

Abstract

A basic model due to Sharp which is useful in the analysis of motorcycle behavior and control is developed. This model is based on linearization of a bicycle model introduced by Whipple, but is augmented with a tire model in which the lateral tire force depends in a dynamic fashion on tire behavior. This model is used to explain some of the important characteristics of motorcycle behavior. The significant dynamic modes exhibited by this model are capsize, weave, and wobble.

Keywords

Bicycle; Capsize; Counter-steering; Motorcycle; Single-track vehicle; Tire model; Weave; Wobble
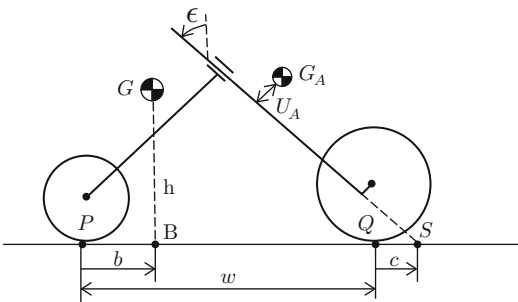
M

Introduction

The bicycle is mankind's ultimate solution to the quest for a human-powered vehicle (Herlihy 2006). The motorcycle just makes riding more fun. Bicycles, motorcycles, scooters, and mopeds are all examples of single-track vehicles and have similar dynamics. The dynamics of a motorcycle are considerably more complicated than that of a four-wheel vehicle such as a car. The first obvious difference in behavior is stability. An unattended upright stationary motorcycle is basically an inverted pendulum and is unstable about its normal upright position, whereas a car has no stability issues in the same configuration. Another difference is that a motorcycle must lean when cornering. Although a car leans a little due to suspension travel, there is no necessity for it to lean in cornering. A perfectly rigid car would not lean. Furthermore, beyond low speeds, the steering behavior of a

motorcycle is not intuitive like that of a car. To turn a car right, the driver simply turns the steering wheel right; on a motorcycle, the rider initially turns the handlebars to the left. This is called counter-steering and is not intuitive.

A Basic Model

To obtain a basic motorcycle model, we start with four rigid bodies: the rear frame (which includes a rigidly attached rigid rider), the front frame (includes handlebars and front forks), the rear wheel, and the front wheel; see Fig. 1. We assume that both frames and wheels have a plane of symmetry which is vertical when the bike is in its nominal upright configuration. The front frame can rotate relative to the rear frame about the steering axis; the steering axis is in the plane of symmetry of each frame and in the nominal upright configuration of the bike, the angle it makes with the vertical is called the rake angle or caster angle and is denoted by ϵ. The rear wheel rotates relative to the rear frame about an axis perpendicular to the rear plane of symmetry and is symmetrical with respect to this axis. The same relationship holds between the front wheel and the front frame. Although each wheel can be three dimensional, we model the wheels as terminating in a knife edge at their boundaries and contact the ground at a single point. Points Q and P are the points on the ground in contact with the front and rear wheels, respectively.

Each of the above four bodies are described by their mass, mass center location, and a 3×3 inertia matrix. Two other important parameters are the wheelbase w and the trail c. The wheelbase is the distance between the contact points of the two wheels in the nominal configuration, and the trail is the distance from the front wheel contact point Q to the intersection S of the steering axis with the ground. The trail is normally positive, that is, Q is behind S. The point G locates the mass center of the complete bike in its nominal configuration, whereas G_A is the location of the mass center of the front assembly (front frame and wheel).

Description of Motion

Considering a right-handed reference frame $e = (\hat{e}_1, \hat{e}_2, \hat{e}_3)$ with origin O fixed in the ground, the bike motion can be described by the location of the rear wheel contact point P relative to O, the orientation of the rear frame relative to e, and the orientation of the front frame relative to the rear frame; see Fig. 2. Assuming the bike is moving along a horizontal plane, the location of P is usually described by Cartesian coordinates x and y. Let reference frame b be fixed in the rear frame with $\hat{b}_1$ and $\hat{b}_3$ in the plane of symmetry with $\hat{b}_1$ along the nominal $P - S$

Motorcycle Dynamics and Control, Fig. 1 Basic model
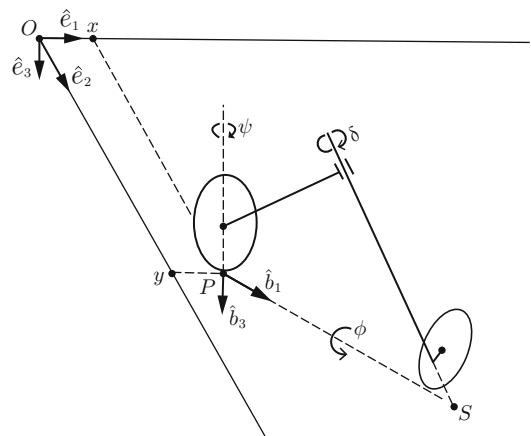
Motorcycle Dynamics and Control, Fig. 2 Description of motion

line; see Fig. 2. Using this reference frame, the orientation of the rear frame is described by a 3-1-2 Euler angle sequence which consists of a yaw rotation by ψ about the 3-axis followed by a lean (roll) rotation by ϕ about the 1-axis and finally by a pitch rotation by θ about the 2-axis. The orientation of the front frame relative to the rear frame can be described by the steer angle δ. Assuming both wheels remain in contact with the ground, the pitch angle θ is not independent; it is uniquely determined by δ and ϕ. In considering small perturbations from the upright nominal configuration, the variation in pitch is usually ignored. Here we consider it to be zero. Also the dynamic behavior of the bike is independent of the coordinates x, y, and ψ. These coordinates can be obtained by integrating the velocity of P and $\dot{\psi}$.

The Whipple Bicycle Model

The "simplest" model which captures all the salient features of a single track vehicle for a basic understanding of low-speed dynamics and control is that originally due to Whipple (1899). We consider the linearized version of this model which is further expounded on in Meijaard et al. (2007). The salient feature of this model is that there is no slip at each wheel. This means that the velocity of the point on the wheel which is instantaneously in contact with the ground is zero; this is illustrated in Fig. 3 for the rear wheel. No slip implies that there is no sideslip which means that the velocity of the wheel contact point ($\bar{v}^P$ in Fig. 3) is parallel to the intersection of the wheel plane with the ground plane; the wheel contact point is the point moving along the ground which is in contact with the wheel.

The rest of this entry is based on linearization of motorcycle dynamics about an equilibrium configuration corresponding to the bike traveling upright in a straight line at constant forward speed $v := v^P$, the speed of the rear wheel contact point P. In the linearized system, the longitudinal dynamics are independent of the lateral dynamics, and in the absence of driving or braking forces, the speed v is constant.

With no sideslip at both wheels, kinematical considerations (see Fig. 4) show that the yaw rate $\dot{\psi}$ is determined by δ; specifically for small angles we have the following linearized relationship:

$$\dot{\psi} = \nu v \delta + \mu \dot{\delta} \qquad (1)$$

where $\nu = c_\epsilon / w$, $c_\epsilon = \cos \epsilon$, and $\mu = c c_\epsilon / w$ is the normalized mechanical trail. In Fig. 4, $\delta_f = c_\epsilon \delta$ is the effective steer angle; it is the angle between the intersections of the front wheel plane and the rear frame plane with the ground. Thus we can completely describe the lateral bike dynamics with the roll angle ϕ and the steer angle δ. To obtain the above relationship, first note that, as a consequence of no sideslip, $\bar{v}^P = v \hat{b}_1$ and $\bar{v}^Q$ is perpendicular to $\hat{f}_2$. Taking the dot product of the expression,
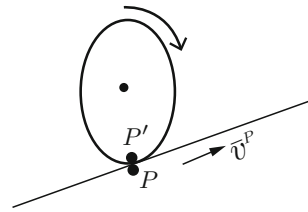
$$\bar{v}^Q = \bar{v}^P + (w+c)\dot{\psi}\, \hat{b}_2 - c(\dot{\psi} + \dot{\delta}_f)\hat{f}_2,$$

with $\hat{f}_2$ while noting that $\hat{b}_1 \cdot \hat{f}_2 = -\sin \delta_f$ and $\hat{b}_2 \cdot \hat{f}_2 = \cos \delta_f$ results in
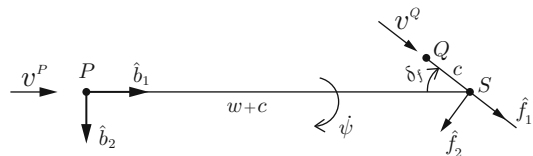
$$0 = -v \sin \delta_f + (w + c)\dot{\psi} \cos \delta_f - c(\dot{\psi} + \dot{\delta}_f).$$

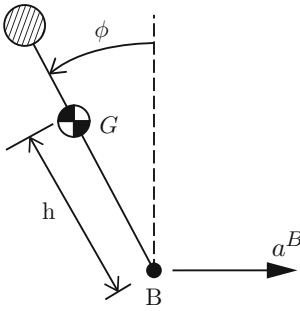Linearization about $\delta = 0$ yields the desired result.

The relationship in (1) also holds for four wheel vehicles. There one can achieve a desired

Motorcycle Dynamics and Control, Fig. 3 No slip: $v^{P'} = 0$

Motorcycle Dynamics and Control, Fig. 4 Some kinematics

Motorcycle Dynamics and Control, Fig. 5 Inverted pendulum with accelerating support point

constant yaw rate $\dot{\psi}_d$ by simply letting the steer angle $\delta = \dot{\psi}_d/vv$. However, as we shall see, a motorcycle with its steering fixed at a constant steer angle is unstable. Neglecting gyroscopic terms, it is simply an inverted pendulum. With its steering free, a motorcycle can be stable over a certain speed range and, if unstable, can be easily stabilized above a very low speed by most people. Trials riders can stabilize a motorcycle at any speed including zero.

To help understand the effect of steer angle on bike behavior, we initially ignore the mass and inertia of the front assembly along with gyroscopic effects, and we assume that the $\hat{b}_1$ axis is a principle axis of inertia of the rear frame with moment of inertia I_{xx}. Angular momentum considerations about the $\hat{b}_1$ axis and linearization results in

$$I_{xx}\ddot{\phi} + mha^B = mgh\phi + (N_f c_\epsilon c)\delta \quad (2)$$

where N_f is the normal force (vertical and upwards) on the front wheel and a^B is the lateral acceleration (perpendicular to rear frame) of point B which is the projection of G onto the $\hat{b}_1$ axis. By considering a moment balance about the pitch axis $\hat{b}_2$ through P, one can obtain that $N_f = mgb/w$. Notice that, with the steering fixed at $\delta = 0$, Eq. (2) is the equation of motion of a simple inverted pendulum whose support axis is accelerating horizontally with acceleration a^B. This is illustrated in Fig. 5.

Basic kinematics reveal that $a^B = v\dot{\psi} + b\ddot{\psi}$ and, recalling relationship (1), Eq. (2) now yields the lean equation:

$$\boxed{I_{xx}\ddot{\phi} - mgh\phi = -m_{\phi\delta}\ddot{\delta} - c_{\phi\delta}v\dot{\delta} - k_{\phi\delta}(v)\delta}$$
$$(3)$$

where $m_{\phi\delta} = \mu mhb > 0$, $c_{\phi\delta} = mh(\mu + bv) > 0$ and $k_{\phi\delta}(v) = -\mu mgb + mhvv^2$. Note that v is a constant parameter corresponding to the nominal speed of the rear wheel contact point.

With $\delta = 0$, we have a system whose behavior is characterized by two real eigenvalues: $\pm\sqrt{mgh/I_{xx}}$. This system is unstable due to the positive eigenvalue $\sqrt{mgh/I_{xx}}$. For v sufficiently large, the coefficient $k_{\phi\delta}(v)$ is positive and one can readily show that the above system can be stabilized with positive feedback $\delta = K\phi$ provided $K > mgh/k_{\phi\delta}(v)$. This helps explain the stabilizing effect of a rider turning the handlebars in the direction the bike is falling. Actually, the rider input is a **steer torque** T_δ about the steer axis.

To explain why an uncontrolled motorcycle can be stable or easily stabilized, one also has to look at the effect that ϕ has on δ; in general, a lean perturbation results in the front assembly turning in the same direction, that is, a positive perturbation of ϕ results in a positive change in δ.

The lean equation also explains why a motorcycle must lean when cornering above a certain speed. Suppose the motorcycle is in a right hand corner of radius R at some constant speed v: in this scenario, $\dot{\psi} = v/R$ and, with δ constant, (1) implies that $\delta = \dot{\psi}/vv = 1/vR$; with δ and ϕ constant, the lean equation now requires that $\phi = k_{\phi\delta}(v)\delta/mgh = k_{\phi\delta}(v)/mghvR$. For higher speeds, $k_{\phi\delta}(v) \approx mhvv^2$; hence $\phi \approx v^2/gR$. Since $a^B = v^2/R$, the lean angle ϕ is approximately a^B/g. Hence, to corner with a lateral acceleration $a^B = v^2/R$, the motorcycle must lean at an angle of approximately a^B/g.

The lean equation can also help explain **counter-steering**; that is, at speeds above a reasonably low speed, one can initiate a turn by turning the handlebars in the opposite direction to which one wants to go; to turn right, one initially turns the handlebars to the left. See Limebeer and Sharp (2006) for further discussion.

Taking into account the mass and inertia of the front assembly, gyroscopic effects and cross products of inertia of the rear frame, one can

show (see Meijaard et al. 2007) that the lean equation (3) still holds with

$$m_{\phi\delta} = \mu I_{xz} + I_{A\epsilon x}$$

$$c_{\phi\delta} = \mu m h + v I_{xz} + \mu S_T + c_\epsilon S_F$$

$$k_{\phi\delta}(v) = k_{0\phi\delta} + k_{2\phi\delta} v^2$$

$$k_{0\phi\delta} = -S_A g,$$

$$k_{2\phi\delta} = v(mh + S_T)$$

Here I_{xx} is the moment of inertia of the total motorcycle and rider in the nominal configuration about the $\hat{b}_1$ axis and I_{xz} is the inertia cross product w.r.t the $\hat{b}_1$ and $\hat{b}_3$ axes. The term $I_{A\epsilon x}$ is the front assembly inertia cross product with respect to the steering axis and the $\hat{b}_1$ axis; see Meijaard et al. (2007) for a further description of this parameter. Also, $S_A = \mu m b + m_A u_A$ where m_A is the mass of the front assembly (front wheel and front frame) and u_A is the offset of the mass center of the front assembly from the steering axis, that is, the distance of this mass center from the steering axis; see Fig. 1. The terms $S_F = I_{Fyy}/r_F$ and $S_T = I_{Ryy}/r_R + I_{Fyy}/r_F$ are gyroscopic terms due the rotation of the front and rear wheels where r_F and r_R are the radii of the front and rear wheels, while I_{Fyy} and I_{Ryy} are the moments of inertias of the front and rear wheels about their axles. It is assumed that the mass center of each wheel is located at its geometric center.

By considering an angular momentum balance about a vertical axis through P, one can obtain an expression for the lateral force at the front wheel. Angular momentum considerations about the steering axis for the front assembly and linearization then yield the steer equation:

$$\boxed{\begin{array}{c} m_{\delta\phi}\ddot{\phi} + m_{\delta\delta}\ddot{\delta} + v c_{\delta\phi}\dot{\phi} + v c_{\delta\delta}\dot{\delta} \\ + k_{\delta\phi}\phi + k_{\delta\delta}(v)\delta = T_\delta \end{array}} \quad (4)$$

where

$$m_{\delta\phi} = m_{\phi\delta}$$

$$m_{\delta\delta} = I_{A\epsilon\epsilon} + 2\mu I_{A\epsilon z} + \mu^2 I_{zz}$$

$$k_{\phi\delta} = k_{\delta\phi}$$

$$k_{\delta\delta}(v) = k_{0\delta\delta} + k_{2\delta\delta} v^2$$

$$k_{0\delta\delta} = -s_\epsilon S_A g,$$

$$k_{2\delta\delta} = v(S_A + s_\epsilon S_F)$$

$$c_{\delta\phi} = -(\mu S_T + c_\lambda S_F)$$

$$c_{\delta\delta} = \mu (S_A + v I_{zz}) + v I_{A\epsilon z}$$

Here, I_{zz} is the moment of inertia of the total motorcycle and the rider in the nominal configuration about the $\hat{b}_3$ axis, $I_{A\epsilon\epsilon}$ is the moment of inertia of the front assembly about the steering axis, and $I_{A\epsilon z}$ is the front assembly inertia cross product with respect to the steering axis and the vertical axis through P. The lean equation (3) combined with the steer equation (4) provide an initial model for motorcycle dynamics. This is a linear model with the nominal speed v as a constant parameter and the rider's steering torque T_δ as an input.

Modes of Whipple Model

At $v = 0$, the linearized Whipple model (3)–(4) has two pairs of real eigenvalues: $\pm p_1, \pm p_2$ with $p_2 > p_1 > 0$; see Fig. 6. The pair $\pm p_1$ roughly describe inverted pendulum behavior of the whole bike with fixed steering, while $\pm p_2$ describe inverted pendulum behavior of the front assembly with the rear frame fixed upright. As v increases the real eigenvalues corresponding to p_1 and p_2 meet and from there on form a
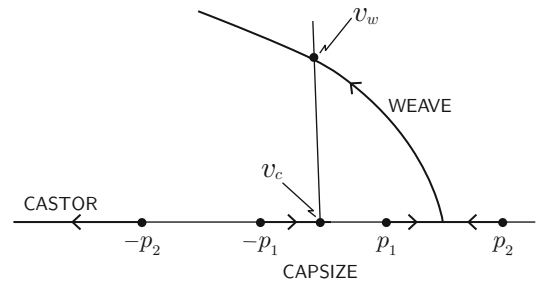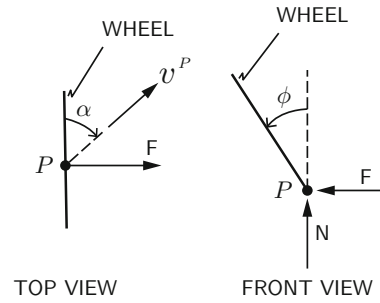

Motorcycle Dynamics and Control, Fig. 6 Variation of eigenvalues of Whipple model with speed v

complex conjugate pair of eigenvalues which result in a single oscillatory mode called the **weave mode**. Initially the weave mode is unstable, but is stable above a certain speed v_w, and for large speeds, its eigenvalues are roughly a linear function of v; thus it becomes more damped and its frequency increases with speed. The eigenvalue corresponding to $-p_2$ remains real and becomes more negative with speed; this is called the **castor** mode, because it roughly corresponds to the front assembly castoring about the steer axis. The eigenvalue corresponding to $-p_1$ also remains real but increases, eventually becoming slightly positive above some speed v_c, resulting in an unstable system; the corresponding mode is called the **capsize** mode. Thus the bike is stable in the autostable speed range (v_w, v_c) and unstable outside this speed range. However, above v_c, the unstable capsize mode is easily stabilized by a rider and usually without conscious effort. This is because the time constant of the unstable capsize mode is very small (Astrom et al. 2005).

Sharp71 Model

The Whipple bicycle model is not applicable at higher speeds. In particular, it does not contain a **wobble mode** which is common to bicycle and motorcycle behavior at higher speeds (Sharp 1971). A wobble mode is characterized mainly by oscillation of the front assembly about the steering axis and can sometimes be unstable. Also, in a real motorcycle, the damping and frequency of the weave mode do not continually increase with speed; the damping usually starts to decrease after a certain speed; sometimes this mode even becomes unstable. At higher speeds, one must depart from the simple non-slipping wheel model. In the Whipple model, the lateral force F on a wheel is simply that force which is necessary to maintain the non-holonomic constraint which requires the velocity of the wheel contact point to be parallel to the wheel plane, that is, no sideslip. Actual tires on wheels slip in the longitudinal and lateral direction, and the lateral force depends on slip in the lateral direction, that is, sideslip. This lateral slip is defined

Motorcycle Dynamics and Control, Fig. 7 Lateral force, slip angle, and camber angle

by the **slip angle** α which is the angle between the contact point velocity and the intersection of the wheel plane and the ground; see Fig. 7.

The lateral force also depends on the tire **camber angle** which is the roll angle of the tire; motorcycle tires can achieve large camber angles in cornering; modern MotoGP racing motorcycles can achieve camber angles of nearly 65°. Thus an initial linear model of a tire lateral force is given by

$$F = N(-k_\alpha \alpha + k_\phi \phi) \qquad (5)$$

where N is the normal force on the tire, $k_\alpha > 0$ is called the tire **cornering stiffness**, and $k_\phi > 0$ is called the **camber stiffness**. Modifying the above Whipple model with the tire force model results in the appearance of the wobble mode. Since lateral forces do not instantaneously respond to changes in slip angle and camber, the dynamic model,

$$\frac{\sigma}{v}\dot{F} + F = N(-k_\alpha \alpha + k_\phi \phi), \qquad (6)$$

is usually used where $\sigma > 0$ is called the **relaxation length** of the tire. This yields more realistic behavior (Sharp 1971). In this model the weave mode damping eventually decreases at higher speeds and the frequency does not continually increase. The frequency of the wobble mode is higher than that of the weave mode and its damping decreases at higher speeds.

Further Models

To obtain nonlinear models, resort is usually made to multi-body simulation codes. In recent years, several researchers have used such codes to make nonlinear models which take into account other features such as frame flexibility, rider models, and aerodynamics; see Cossalter (2006), Cossalter and Lot (2002), Sharp and Limebeer (2001), and Sharp et al. (2004). The nonlinear behavior of the tires is usually modeled with a version of the Magic formula; see Pacejka (2006), Sharp et al. (2004), and Cossalter et al. (2003). Another line of research is to use some of these models to obtain optimal trajectories for high performance; see Saccon et al. (2012).

Summary and Future Directions

We have presented a basic linearized model of a motorcycle or bicycle useful for the understanding and control of these two wheeled vehicles. It seems that inclusion of further features in the model and the consideration of full nonlinear behavior require the use of multibody simulation software. Future research will consider models which will include the engine, transmission, and an active pilot. Autonomous control of these vehicles will also be considered.

Cross-References

▶ Pilot-Vehicle System Modeling
▶ Transmission
▶ Vehicle Dynamics Control

Bibliography

Astrom KJ, Klein RE, Lennarstsson A (2005) Bicycle dynamics and control. IEEE Control Syst Mag 25(4):26–47

Cossalter V, Lot R (2002) A motorcycle multi-body model for real time simulations based on the natural coordinates approach. Veh Syst Dyn 37(6):423–447

Cossalter V, Doria A, Lot R, Ruffo N, Salvador M (2003) Dynamic properties of motorcycle and scooter tires: measurement and comparison. Veh Syst Dyn 39(5):329–352

Cossalter V (2006) Motorcycle dynamics. Second English Edition, LULU

Herlihy DV (2006) Bicycle: the history. Yale University Press, New Haven

Limebeer DJN, Sharp RS (2006) Bicycles, motorcycles, and models. IEEE Control Syst Mag 26(5):34–61

Meijaard JP, Papadopoulos JM, Ruina A, Schwab AL (2007) Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review – including appendix. Proc R Soc A 463(2084):1955–1982

Pacejka HB (2006) Tire and vehicle dynamics, 2nd edn. SAE International, Warrendale, PA

Saccon A, Hauser J, Beghi A (2012) Trajectory exploration of a rigid motorcycle model. IEEE Trans Control Syst Technol 20(2):424–437

Sharp RS (1971) The stability and control of motorcycles. J Mech Eng Sci 13(5):316–329

Sharp RS, Limebeer DJN (2001) A motorcycle model for stability and control analysis. Multibody Syst Dyn 6:123–142

Sharp RS, Evangelou S, Limebeer DJN (2004) Advances in the modelling of motorcycle dynamics. Multibody Syst Dyn 12(3):251–283

Whipple FJW (1899) The stability of the motion of a bicycle. Q J Pure Appl Math 30:312–348

M

Moving Horizon Estimation

James B. Rawlings
University of Wisconsin, Madison,
WI, USA

Synonyms

MHE

Abstract

Moving horizon estimation (MHE) is a state estimation method that is particularly useful for nonlinear or constrained dynamic systems for which few general methods with established properties are available. This entry explains the concept of full information estimation and introduces moving horizon estimation as a computable approximation of full information. The basic design

methods for ensuring stability of MHE are presented. The relationships of full information and MHE to other state estimation methods such as Kalman filtering and statistical sampling are discussed.

Keywords

Full information estimation; Kalman filtering; Statistical sampling

Introduction

In state estimation, we consider a dynamic system from which measurements are available. In discrete time, the system description is

$$x^+ = f(x, w) \qquad y = h(x) + v \qquad (1)$$

The state of the systems is $x \in \mathbb{R}^n$, the measurement is $y \in \mathbb{R}^P$, and the notation x^+ means x at the next sample time. A control input u may be included in the model, but it is considered a known variable, and its inclusion is irrelevant to state estimation, so we suppress it in the model under consideration here. We receive measurement y from the sensor, but the process disturbance, $w \in \mathbb{R}^g$; measurement disturbance $v \in \mathbb{R}^p$; and system initial state, $x(0)$, are considered unknown variables.

The goal of state estimation is to construct or estimate the trajectory of x from only the measurements y. Note that for control purposes, we are usually interested in the estimate of the state at the current time, T, rather than the entire trajectory over the time interval $[0, T]$. In the moving horizon estimation (MHE) method, we use optimization to achieve this goal. We have two sources of error: the state transition is affected by an unknown process disturbance (or noise), w, and the measurement process is affected by another disturbance, v. In the MHE approach, we formulate the optimization objective to minimize the size of these errors thus finding a trajectory of the state that comes close to satisfying the (error-free) model while still fitting the measurements.

First, we define some notation necessary to distinguish the system variables from the estimator variables. We have already introduced the system variables (x, w, y, v). In the estimator optimization problem, these have corresponding decision variables, which we denote by the Greek letters $(\chi, \omega, \eta, \nu)$. The relationships between these variables are

$$\chi^+ = f(\chi, \omega) \qquad y = h(\chi) + \nu \qquad (2)$$

and they are depicted in Fig. 1. Notice that ν measures the gap between the model prediction $\eta = h(\chi)$ and the measurement y. The *optimal* decision variables are denoted $(\hat{x}, \hat{w}, \hat{y}, \hat{v})$, and these optimal decisions are the estimates provided by the state estimator.

Full Information Estimation

The full information objective function is

$$V_T(\chi(0), \boldsymbol{\omega}) = \ell_x\big(\chi(0) - \overline{x}_0\big) + \sum_{i=0}^{T-1} \ell_i(\omega(i), \nu(i)) \tag{3}$$

subject to (2) in which T is the current time, $\boldsymbol{\omega}$ is the estimated sequence of process disturbances, $(\omega(0), \ldots, \omega(T-1))$, $y(i)$ is the measurement at time i, and $\overline{x}_0$ is the prior, i.e., available, value of the initial state. *Full information* here means that we use *all* the data on time interval $[0, T]$ to estimate the state (or state trajectory) at time T. The stage cost $\ell_i(\omega, \nu)$ costs the model disturbance and the fitting error, the two error sources that we reconcile in all state estimation problems.

The full information estimator is then defined as the solution to

$$\min_{\chi(0), \boldsymbol{\omega}} V_T(\chi(0), \boldsymbol{\omega}) \tag{4}$$

The solution to the optimization exists for all $T \in \mathbb{I}_{\geq 0}$ under mild continuity assumptions and choice of stage cost. Many choices of (positive, continuous) stage costs $\ell_x(\cdot)$ and $\ell_i(\cdot)$ are possible, providing a rich class of estimation problems
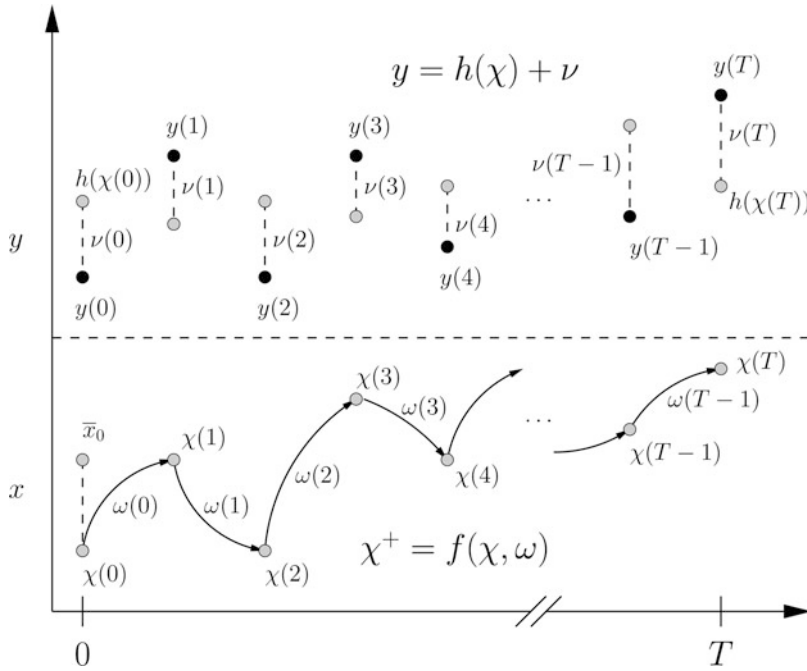
Moving Horizon Estimation, Fig. 1 The state, measured output, and disturbance variables appearing in the state estimation optimization problem. The state trajectory (*gray circles in lower half*) is to be reconstructed given the measurements (*black circles in upper half*)

that can be tailored to different applications. Because the system model (1) and cost function (3) are so general, it is perhaps best to start off by specializing them to see the connection to some classic results.

Related Problem: The Kalman Filter

If we specialize to the linear dynamic model $f(x, w) = Ax + Gw$, $h(x) = Cx$, and let $x(0)$, w, and v be independent, normally distributed *random* variables, the classic Kalman filter is known to be the statistically optimal estimator, i.e., the Kalman filter produces the state estimate that maximizes the conditional probability of $x(T)$ given $y(0), \ldots, y(T)$. The full information estimator is *equivalent* to the Kalman filter given the linear model assumption and the following choice quadratic of stage costs

$$\ell_x(\chi(0), \overline{x}_0) = (1/2) \|\chi(0) - \overline{x}_0\|_{P_0^{-1}}^2$$

$$\ell_i(\omega, v) = (1/2)\left(\|\omega\|_{Q^{-1}}^2 + \|v\|_{R^{-1}}^2 \right)$$

in which random variable $x(0)$ is assumed to have mean $\overline{x}_0$ and variance P_0 and random variables w and v are assumed zero mean with variances Q and R, respectively. The Kalman filter is also a *recursive* solution to the state estimation problem so that only the current mean $\hat{x}$ and variance P of the conditional density are required to be stored, instead of the entire history of measurements $y(i), i = 0, \ldots, T$. This computational efficiency is critical for success in online application for processes with short time scales requiring fast processing.

But if we consider nonlinear models, the maximization of conditional density is usually an intractable problem, especially in online applications. So, MHE becomes a natural alternative for nonlinear models or if an application calls for hard constraints to be imposed on the estimated variables.

Moving the Horizon

An obvious problem with solving the full information optimization problem is that the number of decision variables grows linearly with time T, which quickly renders the problem intractable for continuous processes that have no final time. A natural alternative to full information is to consider instead a finite moving horizon of the most recent N measurements. Figure 2 displays this idea. The initial condition $\chi(0)$ is now replaced by the initial state in the horizon, $\chi(T - N)$, and the decision variable sequence of process disturbances is now just the last N variables $\boldsymbol{\omega} = (\omega(T - N), \ldots, \omega(T - 1))$. Now, the big question remaining is what to do about the neglected, past data. This question is strongly related to what penalty to use on the initial state in the horizon $\chi(T - N)$. If we make this initial state a free variable, that is equivalent to completely discounting the past data. If we wish to retain some of the influence of the past data and keep the moving horizon estimation problem close to the full information problem, then we must choose an appropriate penalty for the initial state. We discuss this problem next.

Arrival Cost. When time is less than or equal to the horizon length, $T \leq N$, we can simply do full information estimation. So we assume throughout that $T > N$. For $T > N$, we express the MHE objective function as
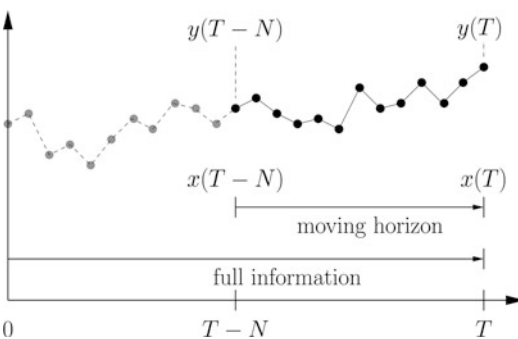
Moving Horizon Estimation, Fig. 2 Schematic of the moving horizon estimation problem

$$\hat{V}_T(\chi(T - N), \boldsymbol{\omega}) = \Gamma_{T-N}(\chi(T - N))$$
$$+ \sum_{i=T-N}^{T-1} \ell_i(\omega(i), v(i))$$

subject to (2). The MHE problem is defined to be

$$\min_{\chi(T-N), \boldsymbol{\omega}} \hat{V}_T(\chi(T - N), \boldsymbol{\omega}) \qquad (5)$$

in which $\boldsymbol{\omega} = \{\omega(T - N), \ldots, \omega(T - 1)\}$ and the hat on V distinguishes the MHE objective function from full information. The designer must now choose this prior weighting $\Gamma_k(\cdot)$ for $k > N$.

To think about how to choose this prior weighting, it is helpful to first think about solving the full information problem by breaking it into *two* non-overlapping sequences of decision variables: the decision variables in the time interval corresponding to the neglected data $(\omega(0), \omega(1), \ldots, \omega(T - N - 1))$ and those in the time interval corresponding to the considered data in the horizon $(\omega(T - N), \ldots, \omega(T - 1))$. If we optimize over the first sequence of variables and store the solution as a function of the terminal state $\chi(T - N)$, we have defined what is known as the arrival cost. This is the optimal cost to arrive at a given state value.

Definition 1 (arrival cost) The (full information) arrival cost is defined for $k \geq 1$ as

$$Z_k(x) = \min_{\chi(0), \boldsymbol{\omega}} V_k(\chi(0), \boldsymbol{\omega}) \qquad (6)$$

subject to (2) and $\chi(k; \chi(0), \boldsymbol{\omega}) = x$.

Notice the terminal constraint that χ at time k ends at value x. Given this arrival cost function, we can then solve the full information problem by optimizing over the remaining decision variables. What we have described is simply the *dynamic programming* strategy for optimizing over a sum of stage costs with a dynamic model (Bertsekas 1995).

We have the following important equivalence.

Lemma 1 (MHE and full information estimation) *The MHE problem* (5) *is equivalent to*

the full information problem (4) *for the choice* $\Gamma_k(\cdot) = Z_k(\cdot)$ *for all* $k > N$ *and* $N \geq 1$.

Using dynamic programming to decompose the full information problem into an MHE problem with an arrival cost penalty is conceptually important to understand the structure of the problem, but it doesn't yet provide us with an implementable estimation strategy because we cannot compute and store the arrival cost when the model is nonlinear or other constraints are present in the problem. But if we are not too worried about the optimality of the estimator and are mainly interested in other properties, such as stability of the estimator, we can find simpler design methods for choosing the weighting $\Gamma_k(\cdot)$. We address this issue next.

Estimator Properties: Stability

An estimator is termed *stable* if small disturbances (w, v) lead to small estimate errors $x - \hat{x}$ as time increases. Precise definitions of this basic idea are available elsewhere (Rawlings and Ji 2012), but this basic notion is sufficient for the purposes of this overview. In applications, properties such as stability and insensitivity to model errors are usually more important than optimality. It is possible for a filter to be *optimal* and still not *stable*. In the linear system context, this cannot happen for "nice" systems. Such nice systems are classified as *detectable*. Again, the precise definition of detectability for the linear case is available in standard references (Kwakernaak and Sivan 1972). Defining detectability for nonlinear systems is a more delicate affair, but useful definitions are becoming available for the nonlinear case as well (Sontag and Wang 1997).

If we lower our sights and do not worry if MHE is equivalent to full information estimation and require only that it be a stable estimator, then the key result is that the prior penalty $\Gamma_k(\cdot)$ need only be chosen *smaller* than the arrival cost as shown in Fig. 3. See Rawlings and Mayne (2009, Theorem 4.20) for a precise statement of this result. Of course this condition includes the flat arrival cost, which does not penalize the initial
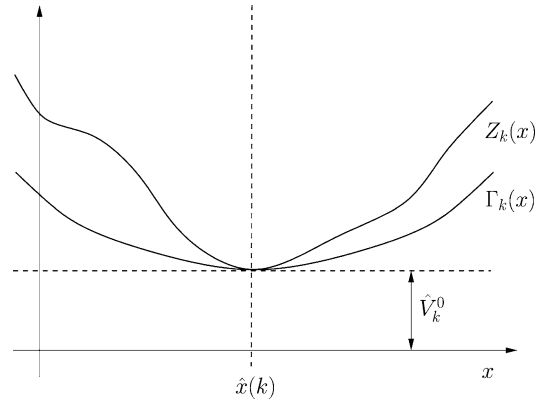
Moving Horizon Estimation, Fig. 3 Arrival cost $Z_k(x)$, underbounding prior weighting $\Gamma_k(x)$, and MHE optimal value $\hat{V}_k^0$; for all x and $k > N$, $Z_k(x) \geq \Gamma_k(x) \geq \hat{V}_k^0$, and $Z_k(\hat{x}(k)) = \Gamma_k(\hat{x}(k)) = \hat{V}_k^0$

state in the horizon at all. So neglecting the past data completely leads to a stable estimator for detectable systems. If we want to improve on this performance, we can increase the prior penalty, and we are guaranteed to remain stable as long as we stay below the upper limit set by the arrival cost.

Related Problem: Statistical Sampling

MHE is based on optimizing an objective function that bears some relationship to the conditional probability of the state (trajectory) given the measurements. As discussed in the section on the Kalman filter, if the system is linear with normally distributed noise, this relationship can be made exact, and MHE is therefore an optimal statistical estimator. But in the nonlinear case, the objective function is chosen with engineering judgment and is only a surrogate for the conditional probability. By contrast, sampling methods such as particle filtering are designed to sample the conditional density also in the nonlinear case. The mean and variance of the samples then provide estimates of the mean and variance of the conditional density of interest. In the limit of infinitely many samples, these methods are exact. The efficiency of the sampling methods depends strongly on the model and the dimension

of the state vector n, however. The efficiency of the sampling strategy is particularly important for online use of state estimators. Rawlings and Bakshi (2006) and Rawlings and Mayne (2009, pp. 329–355) provide some comparisons of particle filtering with MHE and also describe some hybrid methods combining MHE and particle filtering.

Summary and Future Directions

MHE is one of few state estimation methods that can be applied to nonlinear models for which properties such as estimator stability can be established (Rao et al. 2003; Rawlings and Mayne 2009). The required online solution of an optimization problem is computationally demanding in some applications but can provide significant benefits in estimator accuracy and rate of convergence (Patwardhan et al. 2012). Current topics for MHE theoretical research include treating bounded rather than convergent disturbances and establishing properties of suboptimal MHE (Rawlings and Ji 2012). The current main focus for MHE applied research involves reducing the online computational complexity to reliably handle challenging large dimensional, nonlinear applications (Kuhl et al. 2011; Lopez-Negrete and Biegler 2012; Zavala and Biegler 2009; Zavala et al. 2008).

Cross-References

- ▶ Bounds on Estimation
- ▶ Estimation, Survey on
- ▶ Extended Kalman Filters
- ▶ Nonlinear Filters
- ▶ Particle Filters

Recommended Reading

Moving horizon estimation has by this point a fairly extensive literature; a recent overview is provided in Rawlings and Mayne (2009, pp. 356–357). The following references provide either (i) general background required to understand MHE theory and its relationship to other methods or (ii) computational methods for solving the real-time MHE optimization problem or (iii) challenging nonlinear applications that demonstrate benefits and probe the current limits of MHE implementations.

Bibliography

Bertsekas DP (1995) Dynamic programming and optimal control, vol 1. Athena Scientific, Belmont

Kuhl P, Diehl M, Kraus T, Schloder JP, Bock HG (2011) A real-time algorithm for moving horizon state and parameter estimation. Comput Chem Eng 35:71–83

Kwakernaak H, Sivan R (1972) Linear optimal control systems. Wiley, New York. ISBN:0-471-51110-2

Lopez-Negrete R, Biegler LT (2012) A moving horizon estimator for processes with multi-rate measurements: a nonlinear programming sensitivity approach. J Process Control 22:677–688

Patwardhan SC, Narasimhan S, Jagadeesan P, Gopaluni B, Shah SL (2012) Nonlinear Bayesian state estimation: a review of recent developments. Control Eng Pract 20:933–953

Rao CV, Rawlings JB, Mayne DQ (2003) Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations. IEEE Trans Autom Control 48(2): 246–258

Rawlings JB, Bakshi BR (2006) Particle filtering and moving horizon estimation. Comput Chem Eng 30:1529–1541

Rawlings JB, Ji L (2012) Optimization-based state estimation: current status and some new results. J Process Control 22:1439–1444

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison, 576 p. ISBN:978-0-9759377-0-9

Sontag ED, Wang Y (1997) Output-to-state stability and detectability of nonlinear systems. Syst Control Lett 29:279–290

Zavala VM, Biegler LT (2009) Optimization-based strategies for the operation of low-density polyethylene tubular reactors: nonlinear model predictive control. Comput Chem Eng 33(10):1735–1746

Zavala VM, Laird CD, Biegler LT (2008) A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. J Process Control 18: 876–884

MPC

- ▶ Model-Predictive Control in Practice

MRAC

▸ Model Reference Adaptive Control

MSPCA

▸ Multiscale Multivariate Statistical Process Control

Multi-domain Modeling and Simulation

Martin Otter
Institute of System Dynamics and Control,
German Aerospace Center (DLR), Wessling,
Germany

Abstract

One starting point for the analysis and design of a control system is the block diagram representation of a plant. Since it is nontrivial to convert a physical model of a plant into a block diagram, this can be performed manually only for small plant models. Based on research from the last 35 years, more and more mature tools are available to achieve this transformation fully automatically. As a result, multi-domain plants, for example, systems with electrical, mechanical, thermal, and fluid parts, can be modeled in a unified way and can be used directly as input–output blocks for control system design. An overview of the basic principles of this approach is given. This provides also the possibility to use nonlinear, multi-domain plant models directly in a controller. Finally, the low-level "Functional Mockup Interface" standard is sketched to exchange multi-domain models between many different modeling and simulation environments.

Keywords

Introduction

Methods and tools for control system analysis and design usually require an input–output block diagram description of the plant to be controlled. Apart from small systems, it is nontrivial to derive such models from first principles of physics. Since a long time, methods and tools are available to construct such models automatically for one domain, for example, a mechanical model, an electronic, or a hydraulic circuit. These domain-specific methods and tools are, however, only of limited use for the modeling of multi-domain systems.

In the dissertation (Elmqvist 1978), a suitable approach for multi-domain, object-oriented modeling has been developed by introducing a modeling language to define models on a high level based on first principles. The resulting DAE (differential-algebraic equation) systems are transformed with proper algorithms automatically in a block diagram description with input and output signals based on ODEs (ordinary differential equations).

In 1978, the computers were not powerful enough to apply this method on larger systems. This changed in the 1990s, and then the technology has been substantially improved, many different modeling languages appeared (and also disappeared), and the technology was introduced in commercial simulation environments.

In Table 1, an overview of the most important standards, languages, and tools in the year 2013 for multi-domain modeling is given:

The Modelica language is a standard from The Modelica Association (Modelica Association 2012). The first version was released in 1997. Also a large free library is provided with about 1,300 model components from many domains. There are several software tools supporting this modeling language and the free Modelica

M

Multi-domain Modeling and Simulation, Table 1 Multi-domain modeling and simulation environments

Tool name	Web (accessed December 2013)
*Environments based on the **Modelica Standard** (https://www.Modelica.org)*	
CyModelica	http://cydesign.com/
Dymola	http://www.dymola.com/
JModelica.org	http://www.jmodelica.org/
LMS Imagine.Lab AMESim	http://www.lmsintl.com/LMS-Imagine-Lab-AMESim
MapleSim	http://www.maplesoft.com/products/maplesim
MWorks	http://en.tongyuan.cc/
OpenModelica	https://openmodelica.org/
SimulationX	http://www.itisim.com/simulationx/
Wolfram SystemModeler	http://www.wolfram.com/system-modeler/
*Environments based on the **VHDL-AMS Standard** (http://www.eda.ora/twiki/bin/view.cai/P10761)*	
ANSYS Simplorer	http://www.ansys.com/Products
Saber	http://www.synopsys.com/Systems/Saber
SMASH	http://www.dolphin.fr/medal/products/smash/smashoverview.php
SystemVision	http://www.mentor.com/products/sm/systemvision
Virtuoso AMS designer	http://www.cadence.com
*Environments with **vendor-specific** multi-domain modeling languages*	
EcosimPro	http://www.ecosimpro.com/
gPROMS	http://www.psenterprise.com/gproms
OpenMAST	http://www.openmast.org/
Simscape	https://www.mathworks.com/products/simscape
*Environments based on the **Bond Graph** Methodology*	
20-sim	http://www.20sim.com/

Standard Library. The examples of this entry are mostly provided from this standard.

The following registered trademarks are referenced:

Registered trademark	Owner of trademark
AMESim	IMAGINE SA
ANSYS	ANSYS Inc.
Dymola	Dassault Systemes AB
EcosimPro	Empresarios Agrupados A.I.E.
gPROMS	Process Systems Enterprise Limited
MATLAB	The MathWorks Inc
Modelica	Modelica Association
Saber	Sabremark Limited partnership
SimulationX	ITI GmbH
Simulink	The MathWorks Inc
SystemVision	Mentor Graphics Corporation
Virtuoso	Cadence Design

- The VHDL-AMS language is a standard from IEEE (IEEE 1076.1-2007 2007), first released in 1999. It is an extension of the widely used VHDL hardware description language. This language is especially used in the electronics community.

- There are several vendor-specific modeling languages, notably Simscape from Math-Works as an extension to Simulink, as well as MAST, the underlying modeling language of Saber (Mantoolh and Vlach 1992). In 2004, MAST was published as OpenMAST under an open source license.

- Bond graphs (see, e.g., Karnopp et al. 2012) are a special graphical notation to define multi-domain systems based on energy flow. It was invented in 1959 by Henry M. Paynter.

In the section "Modeling Language Principles", the principles of multi-domain modeling based on a modeling language are summarized. In the section "Models for Control Systems", it is shown how such models can be used not only for simulation but also as components in nonlinear control systems. Finally, in the section

"The Functional Mockup Interface", an overview about a low-level standard for the exchange of multi-domain systems is described.

Modeling Language Principles

Schematics: The Graphical View

Modelers nowadays require a simple to use graphical environment to build up models. With very few exceptions, multi-domain environments define models by schematic diagrams. A typical example is given in Fig. 1, showing a simple direct-current electrical motor in Modelica.

In the lower left part, the electrical circuit diagram of the DC motor is visible, consisting mainly of the armature resistance and inductance of the motor, a voltage source, and component "emf" to model in an idealized way the electromotoric forces in the air gap. On the lower right part, the motor inertia, a gear box, and a load inertia are present. In the upper part, the heat transfer of the resistor losses to the environment is modeled with lumped elements.

A component, like a resistor, rotational inertia, or convective heat transfer, is shown as an icon in the diagram. On the border of a component, small rectangular or circular signs are present representing the "physical ports." Ports are connected by lines and model the (idealized) physical or signal interaction between ports of different components, for example, the flow of electrical current or heat or the rigid mechanical coupling.

Components are built up hierarchically from other components. On the lowest level, components are described textually with the respective modeling language (see section "Component Equations").

Coupling Components by Ports

The ports define how of a component can interact with other components. A port contains (a) a definition of the variables that describe the interface and (b) defines in which way a tool can automatically construct the equations of connections. A typical scenario is shown in Fig. 2 where the ports of the three components A, B, C are connected together at one point P:

M
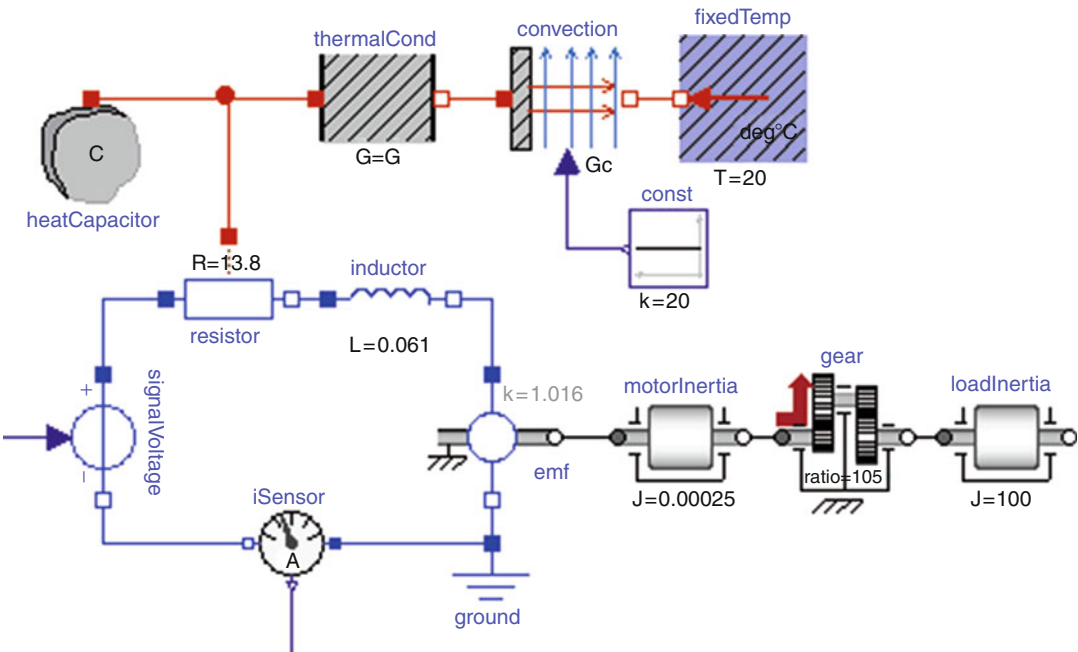
Multi-domain Modeling and Simulation, Fig. 1 Modelica schematic of DC motor with mechanical load and heat losses

When cutting away the connection lines, the resulting system consists of three decoupled components A, B, C and a new component around P describing the infinitesimally small connection point. The balance equations and the boundary conditions of the respective domain must hold at all these components. When drawing the connection lines, enough information must be available in the port definitions so that the tool can construct the equations of the infinitesimally small connection points automatically.

To summarize, the component developer is responsible that the balance equations and boundary conditions are fulfilled for every component (A, B, C in Fig. 2), and the tool is responsible that the balance equations and boundary conditions are also fulfilled at the points where the components are connected together (P in Fig. 2). As a
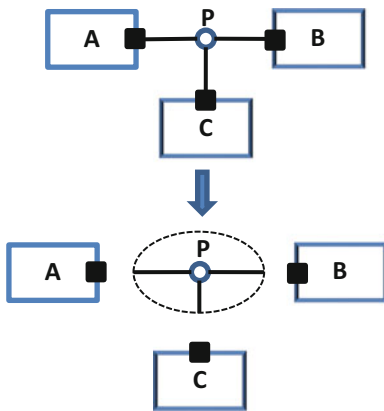
consequence, the balance equations and boundary conditions are fulfilled in the overall model containing all components and all connections.

In order that a tool can automatically construct the equations at a connection point, every port variable needs to be associated to a port variable type. In Table 2, some port variable types of Modelica are shown. In this table it is assumed that $u_1, u_2, \ldots u_n, y, v_1, v_2, \ldots, v_n, f_1, f_2, \ldots, f_n, s_1, s_2, \ldots, s_n$ are corresponding port variables from different components that are connected together at the same point P.

Port variable types "input" and "output" define the "usual" signal connections in block diagrams.

"Potential variables" and "flow variables" are used to define standard physical connections. For example, an electrical port contains the electrical potential and the electrical current at the port, and when connecting electrical ports together, the electrical potentials are identical and the sum of the electrical currents is zero, according to Table 2. This corresponds exactly to Kirchhoff's voltage and current laws.

"Stream variables" are used to describe the connection semantics of intensive quantities in bidirectional fluid flow, such as specific enthalpy or mass fraction. Here, the idealized balance equation at a connection point states, for example, that the sum of the port enthalpy flow rates is zero and the port enthalpy flow rate is computed as the product of the mass flow rate (a flow variable f_i) and the directional specific enthalpy s_i, which is either the (yet unknown) mixing-specific enthalpy s_{mix} when the flow is from the connection point to the port or the specific enthalpy s_i in the port when the flow is from the port to the connection point. More details

Multi-domain Modeling and Simulation, Fig. 2 Cutting the connections around the connection point P results in three decoupled components *A*, *B*, *C* and a new component around *P* describing the infinitesimally small connection point

Multi-domain Modeling and Simulation, Table 2 Some port variable types in Modelica

Port variable type	Connection semantics
Input variables u_i, output variable y	$u_1 = u_2 = \ldots = u_n = y$ (exactly one output variable can be connected to n input variables)
Potential variables v_i	$v_1 = v_2 = \ldots = v_n$
Flow variables f_i	$0 = \sum f_i$
Stream variables s_i (with associated flow variables f_i)	$0 = \sum f_i \hat{s}_i;\ \hat{s}_i = \begin{cases} s_{mix} & \text{if } f_i > 0 \\ s_i & \text{if } f_i \leq 0 \end{cases}$ $(0 = \sum f_i)$

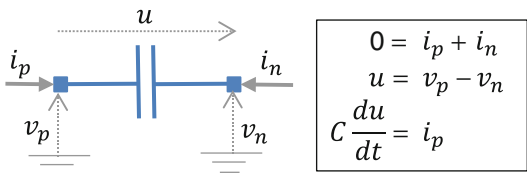**Multi-domain Modeling and Simulation, Table 3** Some port definitions from Modelica

Domain	Port variables
Electrical analog	Electrical potential in [V] *(pot.)* electrical current in [A] *(flow)*
Elec. multiphase	Vector of electrical ports
Electrical quasi-stationary	Complex elec. potential *(pot.)* complex elec. current *(flow)*
Magnetic flux tubes	Magnetic potential in [A] *(pot.)* magnetic flux in [Wb] *(flow)*
Translational (**1**-dim. mechanics)	Distance in [m] *(pot.)* cut-force in [N] *(flow)*
Rotational (**1**-dim. mechanics)	Absolute angle in [rad] *(pot.)* cut-torque in [Nm] *(flow)*
2-dim. mechanics	Position in x-direction in [m] *(pot.)* position in y-direction in [m] *(pot.)* absolute angle in [rad] *(pot.)* cut-force in x-direction in [N] *(flow)* cut-force in y-direction in [N] *(flow)* cut-torque in z-direc. in [Nm] *(flow)*
3-dim. mechanics	Position vector in [m] *(pot.)* transformation matrix in [**1**] *(pot.)* cut-force vector in [N] *(flow)* cut-torque vector in [Nm] *(flow)*
1-dim. heat transfer	Temperature in [K] *(pot.)* heat flow rate in [W] *(flow)*
1-dim. thermo-fluid pipe flow	Pressure in [Pa] *(pot.)* mass flow rate in [kg/s] *(flow)* spec. enthalpy in [J/kg] *(stream)* mass fractions in [**1**] *(stream)*

and explanations are available from Franke et al. (2009). In Table 3 some of the port definitions are shown that are defined in the Modelica Standard Library.

Component Equations

Implementing a component in a modeling language means to (a) define the ports of the component and (b) provide the equations describing the relationships between the port variables. For example, an electrical capacitor with constant capacitance C can be defined by the equations in the right side of Fig. 3.

Such a component has two ports, the pins "p" and "n," and the port variables are the electrical currents i_p, i_n flowing into the respective ports and the electrical potentials v_p, v_n at the ports. The first component equation states that if the current i_p at port "p" is positive, then the current i_n at port "n" is negative (therefore, the current flowing into "p" is flowing out of "n").

$$0 = i_p + i_n$$
$$u = v_p - v_n$$
$$C\frac{du}{dt} = i_p$$

Multi-domain Modeling and Simulation, Fig. 3 Equations of a capacitor component

Furthermore, the two remaining equations state that the derivative of the difference of the port potentials is proportional to the current flowing into port "p."

One important question is how many equations are needed to describe such a component? For an input–output block, this is simple: all input variables are known, and for all other variables, one equation per unknown is needed. Counting equations for physical components, such as a capacitor, is more involved: the requirement

that any type of component connections shall always result in identical numbers of unknowns and equations of the overall system leads to the following counting rule (for a proof, see Olsson et al. 2008):

1. The number of potential and the number of flow variables in a port must be identical.
2. Input variables and variables that appear differentiated are treated as known variables.
3. The number of equations of a component must be equal to the number of unknowns minus the number of flow variables.

In the example of the capacitor, there are 5 unknowns (i_p, i_n, v_p, v_n, du/dt) and **2** flow variables (i_p, i_n). Therefore, $5 - 2 = 3$ equations are needed to define this component.

Modeling languages are used to provide a textual description of the ports and of the equations in a specific syntax. For example, in Modelica the capacitor from Fig. 3 can be defined as Fig. 4 (keywords of the Modelica language are written in boldface):

In VHDL-AMS the capacitor model can be defined as shown in Fig. 5.

One difference between Modelica and VHDL-AMS is that in Modelica all equations need to be explicitly given and port variables (such as p.i) can be directly accessed in the model (Fig. 4). In-stead, in VHDL-AMS (and some other modeling languages), port variables cannot be accessed in a model, and instead via the "**quantity** .. **across** .. **through** .. **to** .." construction, the relationships between the port variables are implicitly defined and correspond to the Modelica equations "0 = p.i + n.i" and "u = p.v − n. v."

Simulation of Multi-domain Systems

Collecting all the component equations of a multi-domain system model together with all connection equations results in a DAE (differential-algebraic equation) system:

$$0 = \mathbf{f}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{w}, \mathbf{y}, \mathbf{u}, t) \tag{1}$$

where $t \in \mathbb{R}$ is time, $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ are variables appearing differentiated, $\mathbf{w}(t) \in \mathbb{R}^{n_w}$ are algebraic variables, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ are outputs, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are inputs, and $\mathbf{f} \in \mathbb{R}^{n_x+n_w+n_y}$ are the DAE equations. Equation (1) can be solved numerically with an integrator for DAE systems; see, for example, Brenan et al. (1996). For DAEs that are linear in their unknowns, a complete theory for solvability is available based on *matrix pencils* (see, e.g., Brenan et al. 1996) and also reliable software for their analysis (Varga 2000).

Unfortunately, only certain classes of *nonlinear* DAEs can be *directly* solved numerically

```
type Voltage = Real (unit="V");
type Current = Real (unit="A");


connector  Pin
         Voltage v;
   flow Current i;
end Pin;


model Capacitor
    parameter Real C(unit="F");
    Pin p,n;
    Voltage u;

equation
    0 = p.i + n.i;
    u = p.v - n.v;
    C*der(u) = p.i;
end Capacitor;
```

Multi-domain Modeling and Simulation, Fig. 4 Modelica model of capacitor component

```
subtype voltage is real;
subtype current is real;
nature electrical is
       voltage across
       current through
       electrical_ref reference;

entity CapacitorInterface IS
   generic(C: real);
   port (terminal p, n: electrical);
end entity CapacitorInterface;

architecture SimpleCapacitor of
            CapacitorInterface is
   quantity u across i through p to n;
begin
   i == C*u'dot;
end architecture SimpleCapacitor;
```

Multi-domain Modeling and Simulation, Fig. 5 VHDL-AMS model of capacitor component

in a reliable way. Domain-specific software, as, e.g., for mechanical systems, transforms the underlying DAE into a form that can be more reliably solved, using domain-specific knowledge. This is performed by differentiating certain equations of the DAE analytically and utilizing special integration methods for the resulting overdetermined set of differential-algebraic equations. Multi-domain simulation software uses the following approaches:

(a) The DAE (1) is directly solved numerically using an implicit integration method, such as a linear multistep method. Typically, all VHDL-AMS simulators use this approach.

(b) The DAE (1) is symbolically transformed in a form that is equivalent to a set of ODEs (ordinary differential equations), and then either explicit or implicit ODE or DAE integration methods are used to numerically solve the transformed system. The transformation is based on the algorithms of Pantelides (1988) and of Mattsson and Söderlind (1993) and might require to analytically differentiate equations. Typically, all Modelica-based simulators, but also EcosimPro, use this approach.

For many models both approaches can be applied successfully. There are, however, systems where approach (a) is successful and fails for (b) or vice versa.

DAEs (1) derived from modeling languages usually have a large number of equations but with only a few unknowns in every equation. In order to solve DAEs of this kind efficiently, both with (a) or (b), typically graph theory and/or sparse matrix methods are utilized. For method (b) the fundamental algorithms have been developed in Elmqvist (1978) and later improved in further publications. For a recent survey and comparison of some of the algorithms, see Frenkel et al. (2012).

Solving the DAE (1) means to solve an initial value problem. In order that this can be performed, a consistent set of initial variables $\dot{\mathbf{x}}_0 = \dot{\mathbf{x}}(t_0)$, $\mathbf{x}_0 = \mathbf{x}(t_0)$, $\mathbf{w}_0 = \mathbf{w}(t_0)$, $\mathbf{y}_0 = \mathbf{y}(t_0)$, $\mathbf{u}_0 = \mathbf{u}(t_0)$ has to be determined first at the initial time t_0. In general, this is a nontrivial task. For example, often (1) shall start in steady state, that is, it is required that $\dot{\mathbf{x}}_0 = 0$ and therefore at the initial time (1) is required to satisfy

$$0 = \mathbf{f}(0, \mathbf{x}_0, \mathbf{w}_0, \mathbf{y}_0, \mathbf{u}_0, t_0) \qquad (2)$$

Equation (2) is a nonlinear algebraic system of equations in the unknowns $\mathbf{x}_0, \mathbf{w}_0, \mathbf{y}_0, \mathbf{u}_0$. These are $n_x + n_w + n_y$ equations for $n_x + n_w + n_y + n_u$ unknowns. Therefore, n_u further conditions must be provided (usually some elements of $\mathbf{u}_0$ and/or $\mathbf{y}_0$ are fixed to desired physical values). Solving (2) for the unknowns is also called "DC operating point calculation" or "trimming." Nonlinear equation solvers are based on iterative methods that require usually a sufficiently accurate initial guess for all unknowns. In a large multi-domain system model, this is not practical, and therefore, methods are needed to solve (2) even if generic guess values in a library are provided that might be far from the solution of the system at hand.

For analog electronic circuit simulations, a large body of theory, algorithms, and software is available to solve (2) based on homotopy methods. The basic idea is to solve a sequence of nonlinear algebraic equation systems by starting with an easy to solve simplified system, characterized by the homotopy parameter $\lambda = 0$. This system is continuously "deformed" until the desired one is reached at $\lambda = 1$. The solution at iteration i is used as guess value for iteration $i + 1$, and at every iteration, the solution is usually computed with a Newton-Raphson method.

The simplest such approach is "source stepping": the initial guess values of all electrical components are set to "zero voltage" and/or "zero current." All (voltage and current) sources start at zero, and their values are gradually increased until the desired source values are reached. This method may not converge, typically due to the severe nonlinearities at switching thresholds in logical circuits.

There are several, more involved approaches, called "probability one homotopy" methods. For these method classes, proofs exist that they converge with probability one (so practically always). These algorithms can only be applied for certain classes of DAEs; see, for example, the

"Variable Stimulus Probability One Homotopy" from Melville et al. (1993).

Although strong results exist for analog electrical circuit simulators, it is difficult to generalize them to the large class of multi-domain systems covered by a modeling language. In Modelica a "homotopy" operator was introduced into the language (Sielemann et al. 2011) in order that a library developer can formulate simple homotopy methods like the "source stepping" in a component library. A generalization of probability one methods for multi-domain systems was developed in the dissertation of Sielemann (2012) and was successfully applied to air distribution systems described as 1-dim. thermo-fluid pipe flow.

Models for Control Systems

Models for Analysis

The multi-domain models from section "Modeling Language Principles" can be utilized to evaluate the properties of a control system by simulation. Also control systems can be designed by nonlinear optimization where at every optimization step one or several simulations of a plant model are executed. Furthermore, modeling environments usually provide a means to linearize the nonlinear DAE (1) of the underlying model around an operating point:

$$\mathbf{x}(t) \approx \mathbf{x}_{op} + \Delta\mathbf{x}(t), \mathbf{w}(t) \approx \mathbf{w}_{op} + \Delta\mathbf{w}(t),$$
$$\mathbf{y}(t) \approx \mathbf{y}_{op} + \Delta\mathbf{y}(t), \ \mathbf{u}(t) \approx \mathbf{u}_{op} + \Delta\mathbf{u}(t)$$
(3)

resulting in

$$\Delta\dot{\mathbf{x}}_{red} = \mathbf{A}\,\Delta\mathbf{x}_{red} + \mathbf{B}\,\Delta\mathbf{u}$$
$$\Delta\mathbf{y} = \mathbf{C}\,\Delta\mathbf{x}_{red} + \mathbf{D}\,\Delta\mathbf{u}$$
(4)

where $\Delta\mathbf{x}_{red}$ is a vector consisting of elements of the vector of $\Delta\mathbf{x}$, the vector $\Delta\mathbf{w}$ is eliminated by exploiting the algebraic constraints, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are constant matrices. Simulation tools provide linear analysis and synthesis methods on this linearized system and/or export it for usage in an environment like Matlab, Maple, Mathematica, or Python.

Multi-domain models might also be used directly in nonlinear Kalman filters, moving horizon estimators, or nonlinear model predictive control. For example, the company ABB is using moving horizon estimation and nonlinear model predictive control based on Modelica models to significantly improve the start-up process of power plants (Franke and Doppelhamer 2006).

Inverse Models

A large body of literature exists about the theory of nonlinear control systems that are based on *inverse* plant models; see, for example, Isidori (1995). Methods such as feedback linearization, nonlinear dynamic inversion, or flat systems use an inverse plant model in the control loop. However, a major obstacle is how to *automatically* utilize an inverse plant model in a controller without being forced to manually set up the equations in the needed form which is not practical for larger systems. Modeling languages can solve this problem as discussed below.

Nonlinear inverse models can be utilized in various ways in a control system. The simplest approach, as feed forward controller, is shown in Fig. 6.

Under the assumption that the models of the plant and of the inverse plant are completely identical and start at the same initial state, then from the construction the control error $\mathbf{e}$ is zero and $\mathbf{y} = \mathbf{T}(s) * \mathbf{y}_{ref}$ where $\mathbf{T}$ is a diagonal matrix with the transfer functions of the low-pass filters on the diagonal (so $\mathbf{y} \approx \mathbf{y}_{ref}$ for reference signals
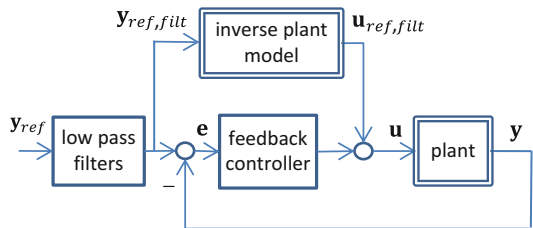
Multi-domain Modeling and Simulation, Fig. 6 Controller with inverse plant model in the feed forward path. The inverse plant model needs usually also derivatives of $\mathbf{y}_{ref}$ as inputs. These derivatives are provided by appropriate low-pass filters

that have a frequency spectrum below the cutoff frequency of the low-pass filters). Since actually the assumption is usually not fulfilled, there will be a nonzero control error **e** and the feedback controller has to cope with it. This controller structure with a nonlinear inverse plant model has the advantage that the feed forward part is useful over the complete operating range of the plant.

Various other structures with nonlinear plant models are discussed in Looye et al. (2005), such as compensation controllers, feedback linearization controllers, and nonlinear disturbance observers.

It turns out that nonlinear inverse plant models can be generated automatically with the techniques that have been developed for modeling languages; see section "Modeling Language Principles". In particular, constructing an inverse model from (1) means that the inputs u are defined to be outputs, so they are no longer knowns but unknowns, and outputs y are defined to be inputs, so they are no longer unknowns but knowns. The resulting system is still a DAE and can therefore be handled as any other DAE.

Therefore, defining an inverse model with a modeling language just requires exchanging the definition of input and output signals. In Modelica, this can be graphically performed with the nonstandard input–output block from Fig. 8.

This block has two inputs and two outputs and described by the equations

$$u1 = u2; \quad y1 = y2$$

From a block diagram point of view this looks strange. However, from a DAE point of view, this just states constraints between two input and two output signals. In Fig. 8, it is shown how this block can be used to invert a simple second order system.

The output of the low-pass filter is connected to the output of the second-order system and therefore this model computes the input of the second-order system, from the input of the filter.

A Modelica environment will generate from this type of definition the inverse model, thereby differentiating equations analytically and solving algebraic variables of the model in a different way as for a simulation model. The whole transformation is nontrivial, but it is just the standard method used by Modelica tools as for any other type of DAE system.

The question arises whether a solution of the inverse model exists, is unique, and whether the model is stable (otherwise, it cannot be applied in a control system). In general, a nonlinear inverse model consists of linear and/or nonlinear algebraic equation systems and of linear and/or nonlinear differential equations. Therefore, from a formal point of view, the same theorems as for a general DAE apply; see, for example, Brenan et al. (1996). Furthermore, all these equations need to be solved with a numerical method. For some classes of systems, it can be shown that mathematically a unique solution exists and that the system is stable. However, in general, one cannot expect that it is possible to provide such a proof for complex inverse plant models. Still, inverse plant models have been successfully utilized by automatic generation from a Modelica tool, e.g., for robots, satellites, aircrafts, vehicles, and thermo-fluid systems.

The Functional Mockup Interface

Many different types of simulation environments are in use. One cannot expect that a generic approach as sketched in section "Modeling Language Principles" will replace all these environments with their rich set of domain-specific knowledge, analysis, and synthesis features. Practically, all simulation environments provide a vendor-specific interface in order that a user can import components that are not describable by the simulation environment itself. Typically, this requires to provide a component as a set of C or Fortran functions with a particular calling interface. In the control community, the most widely used approach of this kind is the S-Function interface from The MathWorks, where Simulink is used as integration platform, and model components from other environments are imported as S-Functions.
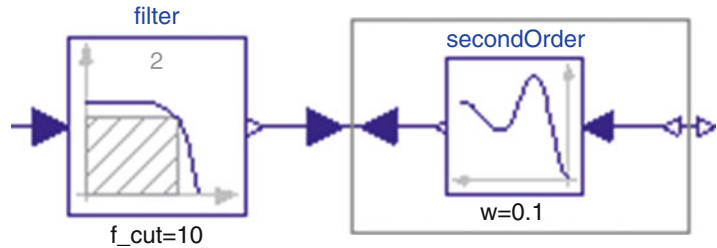
Multi-domain Modeling and Simulation, Fig. 7 Modelica InverseBlockConstraint block
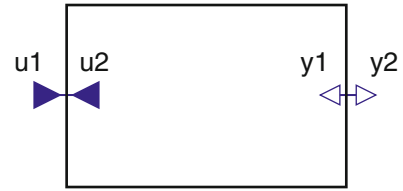
Multi-domain Modeling and Simulation, Fig. 8 Inversion of a second-order system in Modelica

In 2010 the vendor-independent standard "Functional Mockup Interface 1.0" was published (FMI Group 2010). This is a low-level standard for the exchange of models between different simulation environments. This standard allows to exchange only either the model equations (called "FMI for Model Exchange") or the model equations with an embedded solver (called "FMI for Co-Simulation"). This standard was quickly adopted by many simulation environments, and in 2013 there are more than 40 tools that support it (for an actual list of tools, see https://www.fmi-standard.org/tools). In particular nearly all Modelica environments can export Modelica models in this format, and therefore, Modelica multi-domain models can be imported in other environments with low effort.

A software component which implements the FMI is called Functional Mockup Unit (FMU). An FMU consists of one zip-file with extension ".fmu" containing all necessary components to utilize the FMU either for Model Exchange, for Co-Simulation, or for both. The following summary is an adapted version from Blochwitz et al. (2012):

1. An *XML-file* contains the definition of all exposed variables of the FMU, as well as other model information. It is then possible to run the FMU on a target system without this information, i.e., without unnecessary overhead. Furthermore, this allows determining all properties of an FMU from a text file, without actually loading and running the FMU.

2. A set of *C-functions* is provided to execute model equations for the Model Exchange case and to simulate the equations for the Co-Simulation case. These C-functions can be provided either in binary form for different platforms or in source code. The different forms can be included in the same model zip-file.

3. Further data can be included in the FMU zip-file, especially a model icon (bitmap file), documentation files, maps and tables needed by the model, and/or all object libraries or DLLs that are utilized.

Summary and Future Directions

Multi-domain modeling based on a DAE description and defined with a modeling language is an established approach, and many tools support it. This allows to conveniently define plant models from many domains for the design and evaluation of control systems. Furthermore, nonlinear inverse plant models can be easily constructed with the same methodology and can be utilized in various ways in nonlinear control systems.

Current research focuses on the support of the complete life cycle: defining requirements of a system formally on a "high level,"

considerably improving testing by checking these requirements automatically when evaluating a system design by simulations, and providing complete tool chains from nonlinear multi-domain models to embedded systems. The latter will allow convenient and fast target code generation of nonlinear controllers, extended and unscented Kalman filters, optimization-based controllers, or moving horizon estimators.

Furthermore, the methodology itself is further improved. For example, in 2012, Modelica was extended with language elements to define multi-rate sampled data systems in a precise way, as well as state machines.

Cross-References

▶ Computer-Aided Control Systems Design: Introduction and Historical Overview
▶ Extended Kalman Filters
▶ Feedback Linearization of Nonlinear Systems
▶ Interactive Environments and Software Tools for CACSD
▶ Model Building for Control System Synthesis

Bibliography

Blochwitz T, Otter M, Akesson J, Arnold M, Clauß C, Elmqvist H, Friedrich M, Junghanns A, Mauss J, Neumerkel D, Olsson H, Viel A (2012) The functional mockup interface 2.0: the standard for tool independent exchange of simulation models. In: Proceedings of 9th international modelica conference, Munich, 3–5 Sept 2012, pp 173–184. http://www.ep.liu.se/ecp/076/017/ecp12076017.pdf

Brenan KE, Campbel SL, Petzold LR (1996) Numerical solution of initial-value problems in differential-algebraic equations. SIAM, Philadelphia

Elmqvist H (1978) A structured model language for large continuous systems. Dissertation. Report CODEN:LUTFD2/(TFRT–1015), Department of Auto Control, Lund Institute of Technology, Lund. http://www.control.lth.se/database/publications/article.pike?action=fulltext&artkey=elm78dis

FMI Group (2010) The Functional Mockup Interface for Model Exchange and for Co-Simulation, version 1.0. https://www.fmi-standard.org

Franke R, Doppelhamer J (2006) Online application of Modelica models in the industrial IT extended automation system 800xA. In: Proceedings of 6th Modelica conference, Vienna, 4–5 Sept 2006, pp 293–302. https://modelica.org/events/modelica2006/Proceedings/sessions/Session3c2.pdf

Franke R, Casella F, Otter M, Sielemann M, Mattsson SE, Olsson H, Elmqvist H (2009) Stream connectors – an extension of modelica for device-oriented modeling of convective transport phenomena. In: Proceedings of 7th Modelica conference, Como, pp 108–121. https://www.modelica.org/events/modelica2009/Proceedings/memorystick/pages/papers/0078/0078.pdf

Frenkel J, Kunze G, Fritzson P (2012) Survey of appropriate matching algorithms for large scale systems of differential algebraic equations. In: Proceedings of the 9th international Modelica conference, Munich, 3–5 Sept 2012, pp 433–442. http://www.ep.liu.se/ecp/076/045/ecp12076045.pdf

IEEE 1076.1-2007 (2007) IEEE standard VHDL analog and mixed-signal extensions. Standard of IEEE. http://standards.ieee.org/findstds/standard/1076.1-2007.html

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York

Karnopp DC, Margolis DL, Rosenberg RC (2012) System dynamics: modeling, simulation, and control of mechatronic systems, 5th edn. Wiley, Hoboken

Looye G, Thümmel M, Kurze M, Otter M, Bals J (2005) Nonlinear inverse models for control. In: Proceedings of the 4th international Modelica conference, Hamburg, 7–8 March 2005, p 267. https://www.modelica.org/events/Conference2005/onlineproceedings/Session3/Session3c3.pdf

Mattsson SE, Söderlind G (1993) Index reduction in differential-algebraic equations using dummy derivatives. SIAM J Sci Comput 14:677–692

Mantoolh HA, Vlach M (1992) Beyond spice with saber and MAST. In: IEEE international symposium on circuits and systems, San Diego, May 10–13 1992, vol 1, pp 77–80

Melville RC, Trajkovic L, Fang SC, Watson LT (1993) Artificial parameter homotopy methods for the DC operating point problem. IEEE Trans Comput Aided Des Integr Circuits Syst 12:861–877. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.212.9327&rep=rep1&type=pdf

Modelica Association (2012) Standard, Modelica transactions on computer – a unified object-oriented language for systems modeling. Language specification, version 3.3. https://www.modelica.org/documents/ModelicaSpec33.pdf

Olsson H, Otter M, Mattsson SE, Elmqvist H (2008) Balanced models in Modelica 3.0 for increased model quality. In: Proceedings of 6th Modelica conference, Bielefeld, pp 21–33. https://modelica.org/events/modelica2008/Proceedings/sessions/session1a3.pdf

Pantelides CC (1988) The consistent initialization of differential-algebraic systems. SIAM J Sci Stat Comput 9:213–233

M

Sieleman M (2012) Device-oriented modeling and simulation in aircraft energy systems design. Dissertation, Dr. Hut. ISBN 3843905045

Sielemann M, Casella F, Otter M, Clauß C, Eborn J, Mattsson SE, Olsson H (2011) Robust initialization of differential-algebraic equations using homotopy. In: 8th international Modelica conference, Dresden. https://www.modelica.org/events/modelica2011/Proceedings/pages/papers/041ID154afv.pdf

Varga A (2000) A descriptor systems toolbox for Matlab. In: Proceedings of CACSD'2000 IEEE international symposium on computer-aided control system design, Anchorage, 25–27 Sept 2000, pp 150–155. http://elib.dlr.de/11629/01/vargacacsd2000p2.pdf

Multiscale Multivariate Statistical Process Control

Julian Morris
School of Chemical Engineering and Advanced Materials, Centre for Process Analytics and Control Technology, Newcastle University, Newcastle Upon Tyne, UK

Synonyms

MSPCA

Abstract

Dynamic processes, both continuous and batch, are characterised by autocorrelated measurements which are allied to the effects of process dynamics and disturbances. The common multivariate statistical process control (MSPC) approaches have been to use principal component analysis (PCA) or projection to latent structures (PLS) to build a model that captures the simultaneous correlations amongst the variables, but that ignores the serial correlation in the data during normal operations. Under such conditions it is difficult to perform efficient fault detection and diagnosis. An alternative approach to account for the process dynamics in MSPC is to use multiresolution analysis (MRA) by way of wavelet decomposition. Here, the individual measurements are decomposed into different

scales (or frequencies) and the signals in each decomposed scale are then used for MSP which provides an indirect way of handling process dynamics.

Keywords

Multiresolution analysis; Partial least squares (PLS); Principal component analysis (PCA); Projection to latent structures (PLS); Wavelet transform

Definition

Multiscale principal component analysis (MSPCA) and its extension multiscale projection to latent structures (MSPLS) combine the abilities of these multivariate tools to decorrelate the variables by extracting linear relationships with that of wavelet analysis, to extract deterministic features and approximately de-correlate autocorrelated measurements. Multiscale modeling makes use of the wavelet transform which allows a signal (measurement) to be viewed in multiple resolutions with each resolution representing a different frequency. That is, wavelet transform allows complex information to be decomposed into basic components at different positions and scales.

Motivation and Background

One of the drawbacks of the conventional PCA (or PLS)-based MSPC is that although the PCA/PLS model captures the correlations among the variables, it ignores the serial (auto)correlation in the process variables and measurements. One way to overcome this issue is to include time-lagged variables in the PCA or PLS model. In this way, PCA and PLS will explicitly model both the correlations among the variables and the serial correlations in the individual variables. The impact is an increase in the number principal components required, but the multivariate monitoring model will be able to detect any changes in the serial correlation of

the variables as well as changes in relationships among the variables. This article focuses on multiscale-multiway PCA using batchwise data unfolding. However, the methodology can equally be applied to PLS-based process performance monitoring (MSPC).

In multivariate statistical process control (MSPC), the multivariate statistical techniques of principle component analysis (PCA) and projection to latent structures {Partial Least Squares} (PLS) together with monitoring metrics based on Hotelling's T^2 (directly related to the Mahalanobis distance that monitors the fit of new observations to the model space) and the squared prediction error (SPE) or Q statistic (that monitors the residual space-model mismatch) are used to simultaneously monitor the process variables (Kourti and MacGregor 1996; Qin 2003). A recent survey provides an excellent state-of-the-art review of the methods and applications of data-driven fault detection and diagnosis that have been developed over the last two decades (Qin 2012).

Process measurements typically exhibit multi-scale behavior as a consequence of representing the cumulative effect of a number of underlying process phenomena including process dynamics, measurement noise, and disturbances. To address these issues, methodologies are required to address (i) the multiscale nature of process data and (ii) the inability of some existing algorithms to handle autocorrelation. One approach is through the use of multiresolution analysis and wavelets Mallat (1998). Informative discussions and application studies related to using multiresolution analysis and wavelet decompositions to enhance PCA-based process monitoring and fault detection have been presented, for example, by Bakshi (1998), Misra et al. (2002), Aradhye et al. (2003), Lu et al. (2003), Yoon and MacGregor (2004) and Reis and Saraiva (2006). Yoon and MacGregor in their comprehensive MSPCA study discussed their approach in the context of other multiscale approaches and illustrated the methodology using simulated data from a continuous stirred-tank reactor system. A major contribution of the paper was to extend fault isolation methods based on contribution plots to multiscale PCA approaches. Although some 9 years old, Ganesan et al. (2004)

provided review of wavelet-based multiscale statistical process monitoring.

The Approach

Multiresolution analysis (MRA) provides the theoretical basis for the derivation of a computationally efficient algorithm for the wavelet transform Mallat (1998). MRA allows the dynamic aspects of the data in to be taken into account in MSPC. The individual signals are decomposed into different scales (frequencies), and data in each decomposed scale are then used for MSPC which provides an indirect approach to handling process dynamics. Multiscale MSPC (MSPCA) enables the simultaneous extraction of process correlations across data as well as accounting for autocorrelation within sensor data. In this way, it captures correlations among the process variables made by various events occurring at different scales.

MSPCA calculates the principal components of wavelet coefficients at each scale and combines these at the relevant wavelet scales. Due to its multiscale nature, MSPCA is very useful for the modeling of data containing contributions from events whose behavior changes over both time and frequency. Process monitoring by MSPCA, and process prediction by MSPLS, involves combining those scales where significant events are detected. Approximate de-correlation of wavelet coefficients also makes MSPCA effective for the monitoring of autocorrelated measurements.

The Algorithm

Wavelets are a family of basis functions that provide a mapping from the time domain to the time-frequency domain. They can be used to decompose the signal into different resolutions by projecting onto the corresponding wavelet basis functions using multiresolution analysis (MRA). A wavelet set is constructed from a fundamental basis function or the mother wavelet by a process of translation and dilation. The wavelet set is defined as wavelet analysis which provides

methodologies for the extraction of the time and frequency content of a signal. Conventional frequency analysis based on the Fourier transform consists of decomposing a signal into sine waves of different frequencies. Wavelet analysis decomposes the original signal in a similar manner. The major difference is that while Fourier analysis uses sine waves of infinite length, multiresolution analysis uses waveforms of finite length. The finite length of the wavelets allows them to describe local events in both the time and frequency domain.

The wavelet transform, an extension to the Fourier transform, projects the original signal down onto wavelet basis functions, providing a mapping from the time domain to the timescale plane. The wavelet functions, which are localized in the time and frequency domain, are obtained from a single prototype wavelet, the *mother wavelet*, by dilation and translation. The wavelet set is defined as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi \left(\frac{t-b}{a} \right)$$

where ψ is the mother wavelet function, a the dilation parameter, and b the translation parameters, and the factor $\frac{1}{\sqrt{|a|}}$ is used to ensure that each wavelet function has the same energy as the mother wavelet. The discrete wavelet transform with dyadic dilation and translation is used in this overview. A definition of continuous and discrete wavelet transforms can be found in Daubechies (1992). In the discrete case, the dilation and translation parameters are discretized as $a = a_0^m$ and $b = kb_0 a_0^m$. If $a_0 = 2$ and $b_0 = 1$, a *dyadic* dilation and translation is carried out; however, a_0 and b_0 are not restricted to these values. The discrete wavelet form, which is widely used in process monitoring and chemical signal analysis, is

$$\psi_{jk}(t) = a_0^{-j/2} \psi(a_0^{-j_t} - kb_0)$$

A recursive algorithm for wavelet decomposition and the reconstruction of a discrete signal of dyadic length is often used Mallat (1998) and is known as the pyramid algorithm. The fast discrete wavelet decomposition consists of

three components, low-pass filters $L(n)$, high-pass filters $H(n)$, and dyadic decimation. By passing the input signal through this pair of filters, the projection of the original signal onto the scaling and wavelet functions for the multiresolution analysis is performed. Dyadic decimation, or down-sampling, removes every odd member of a sequence, thus halving the original number of samples. The low-pass filter resembles a moving average, while the high-pass filter extracts the detailed information contained in the signal. The discrete wavelet transform operates by taking a sequence of values, applying $L(n)$ and $H(n)$ and then repeating this same procedure to the approximation coefficients. In this way, the original signal vector is *smoothed and halved* through L, and the vector of approximation coefficients is again *smoothed and halved* through L. Successive application of the low-pass filter results in the approximation coefficients, becoming an increasingly smooth version of the original signal. At the same time as smoothing the signal, each iteration extracts the high frequency information in the data. The repeated application of L, followed by H is, in effect, a band-pass filter. The result of applying high-/low-pass filters to a signal is a set of coefficients describing the details of the signals $\mathbf{D}_L$ and a second set describing the approximations of the signals $\mathbf{A}_L$. The original signal s can then be represented by

$$x(t) = \sum_{j=1}^{L} D_j(t) + A_L(t)$$

where D_j and A_j are referred to as the jth level wavelet details and approximation, respectively.

Figure 1 shows schematically the multiresolution-based wavelet decomposition.

One of the most popular choices of wavelets are those of the Daubechies' family. These wavelets are compactly supported in the time domain and have good frequency domain decay. Moreover, Daubechies' wavelets (DaubN) possess a different type of smoothness which is determined by the vanishing moments N. This makes it possible to match the wavelet smoothness to the smoothness of the signals to be analyzed. The signal can then be decomposed

Multiscale Multivariate Statistical Process Control, Fig. 1 Schematic of multiresolution wavelet decomposition
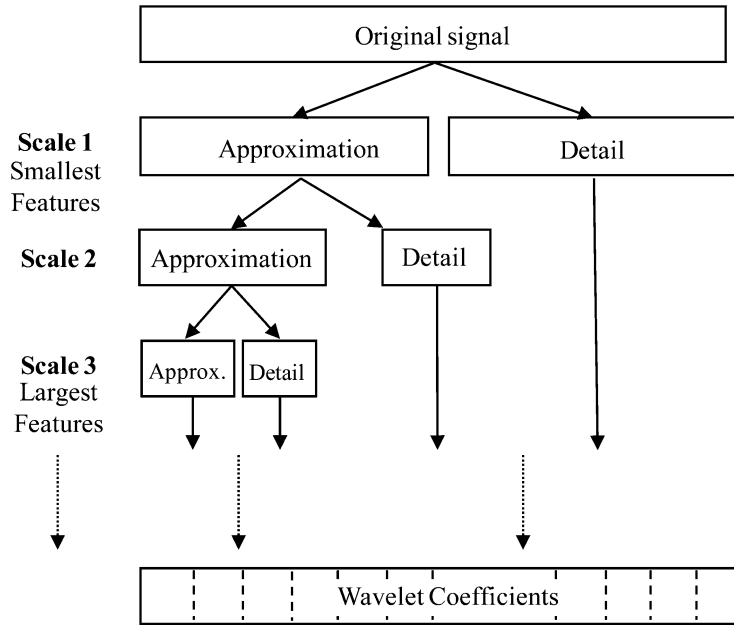
into its contributions from multiple scales as a weighted sum of dyadically discretized orthonormal basis functions:

$$x(t) = \sum_{m=1}^{L} \sum_{k=1}^{N} d_{mk} \psi_{mk}(t) + \sum_{k=1}^{N} a_{lk} \varphi_{lk}(t)$$

where $x(t)$ represents the process measurements, d_{mk} represents the wavelet or detail signal coefficient at scale m and location k, and a_{Lk} represent the scaled signal or scaling function coefficient of $\phi(t)$ at the coarsest scale L and location k. The scaling function, or father wavelet, ϕ_{mk} captures the low-frequency content of the original signal that is not captured by wavelets at the corresponding or finer scales.

The wavelet transformation is applied to decompose a multivariate signal $\mathbf{X}$ into its approximate, $\mathbf{A}_1$ to $\mathbf{A}_L$, and detail, $\mathbf{D}_1$ to $\mathbf{D}_L$, coefficients for the first to Lth level, respectively. For more information, see Bakshi (1998), Misra et al. (2002) and Aradhye et al. (2003). Figure 2 shows a schematic representation of a typical MSPCA multivariate statistical process control scheme.

An example of the application of multiway-multiscale MPCA to a benchmark-fed batch

fermentation process (Birol et al. http://www.chee.iit.edu~control/software.html) was presented by Alawi and Morris (2007). The application used a combination of multiblock statistical modeling approaches together with multiscale-multiway batch monitoring. Figure 3 shows the multiscale-multiway monitoring scheme for process monitoring and fault detection. At every time point, the batch process variables are decomposed into scales to the wavelet domain and then reconstructed back to the time domain. The scales/details and the approximations are collected into separate matrices (blocks). Multiblock PCA is then applied to the wavelets details and approximation. Fault detection based on the T_s^2 and Q_s statistics was used along with contribution plots incorporating confidence bounds to enhance fault diagnosis.

Figure 4 compares the monitoring statistics for the multiscale-multiway PCA and conventional multiway PCA for a slowly drifting sensor fault showing the potential for multiscale MPCA (MSPCA) in being able to detect faster subtle process and sensor faults than conventional multiway MSPC. It is noted that sensor drift is confined to one scale band at low frequency. It has been observed that multiscale approaches appear to provide little improvement if a fault effect is
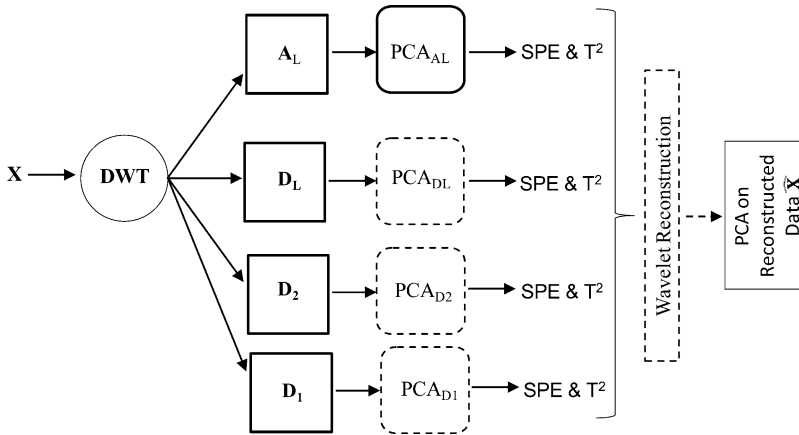
Multiscale Multivariate Statistical Process Control, Fig. 2 Schematic representation of multiscale PCA-based MSPC
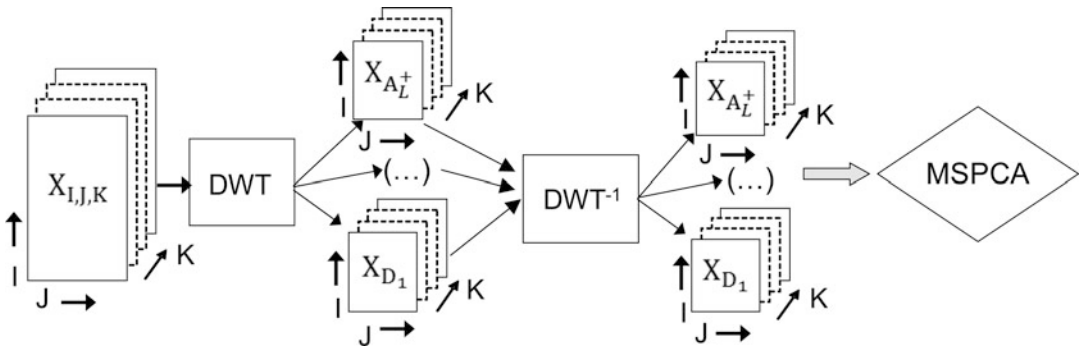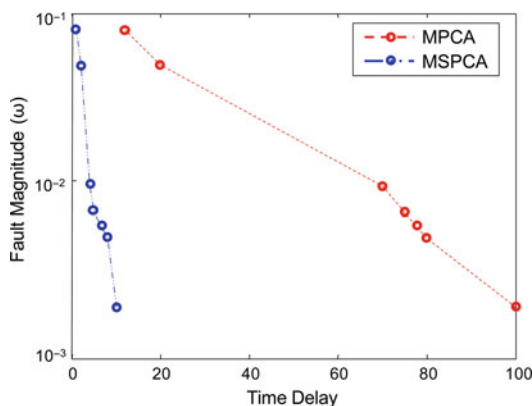

Multiscale Multivariate Statistical Process Control, Fig. 3 Multiscale-multiway batch process monitoring scheme

spread over more than one frequency band or the fault effect occurs mainly in a scale with dominant variance. Thus, a monitoring method that gives the best detection and identification of faults will depend on the fault characteristics with multiscale approaches, providing an advantage when the faults localized in frequency or that appear in scales that normally have small variance.

Other Applications of Multiscale MPCA

There have a number of nonlinear extensions. For example, multiscale PLS approaches have been developed, e.g., Teppola and Minkkinen (2000) and Lee et al. (2009). Nonlinear approaches have also been explored. For example, Lee et al. (2004) proposed a batch monitoring approach using multiway kernel principal component analysis, Shao et al. (1999) proposed a wavelet-based nonlinear PCA algorithm, Choi et al. (2008) described a study of a kernel-based MSPCA algorithm for nonlinear multiscale monitoring, and most recently Zhang and Ma (2011) compared fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS. Wavelet multiscale approaches have also been widely discussed in spectroscopic data processing (Shao et al. 2004).

Lee J-M, Yoo C-K, Lee I-B (2004) Fault detection of batch processes using multiway Kernel principal component analysis. Comput Chem Eng 28(9):1837–1847

Lee HW, Lee MW, Park JM (2009) Multi-scale extension of PLS algorithm for advanced on-line process monitoring. Chemom Intell Lab Syst 98:201–212

Liu Z, Cai W, Shao X (2009) A weighted multiscale regression for multivariate calibration of near infrared spectra. Analyst 134:261–266

Lu N, Wang F, Gao F (2003) Combination method of principal component and wavelet analysis for multivariate process monitoring and fault diagnosis. Ind Eng Chem Res 42:4198–4207

Mallat SG (1998) Multiresolution approximations and wavelet orthonormal bases. Trans Am Math Soc 315:69–87

Misra MH, Yue H, Qin SJ, Ling C (2002) Multivariate process monitoring and fault diagnosis by multi-scale PCA. Comp Chem Eng 26:1281–1293

Qin SJ (2003) Statistical process monitoring: basics and beyond. J Chemom 17:480–502

Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis. Annu Rev Control 36:220–234

Reis MS, Saraiva PM (2006) Multiscale statistical process control with multiresolution data. AIChE J 52:2107–2119

Shao R, Jia F, Martin EB, Morris AJ (1999) Wavelets and nonlinear principal components analysis for process monitoring. Control Eng Pract 7:865–879

Shao X-G, Leung AK-M, Chau F-T (2004) Wavelet: a new trend in chemistry. Acc Chem Res 36:276–283

Teppola P, Minkkinen P (2000) Wavelet-PLS regression models for both exploratory data analysis and process monitoring. J Chemom 14:383–399

Yoon S, MacGregor JF (2004) Principal-component analysis of multiscale data for process monitoring and fault diagnosis. AIChE J 50(11):2891–2903

Zhang Y, Ma C (2011) Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS. Chem Eng Sci 66:64–72

Multiscale Multivariate Statistical Process Control, Fig. 4 Comparison of MPCA and multiscale MSPCA for a range of subtle sensor drift faults magnitude ω against fault detection delay

Cross-References

▶ Controller Performance Monitoring
▶ Fault Detection and Diagnosis
▶ Statistical Process Control in Manufacturing

M

Bibliography

Alawi A, Zhang J, Morris AJ (2007) Multiscale Multiblock Batch Monitoring: Sensor and Process Drift and Degradation. DOI: 10.1021/op400337x April 25 2014

Aradhye HB, Bakshi BR, Strauss A, Davis JF (2003) Multiscale SPC using wavelets: theoretical analysis and properties. AIChE J 49(4):939–958

Bakshi BR (1998) Multiscale PCA with application to multivariate statistical process monitoring. AIChE J 44:1596–1610

Birol G, Undey C, Cinar AA (2002) Modular simulation package for fed-batch fermentation: penicillin production. Comput Chem Eng 26:1553–1565

Choi SW, Morris J, Lee I-B (2008) Nonlinear multiscale modelling for fault detection and identification. Chem Eng Sci 63:2252–2266

Daubechies I (1992) Ten lectures on wavelets. SIAM, Philadelphia

Ganesan R, Das TT, Venkataraman V (2004) Wavelet-based multiscale statistical process monitoring: a literature review. IIE Trans 36:787–806

Kourti T, MacGregor JF (1996) Multivariate SPC methods for process and product monitoring. J Qual Technol 28:409–428

Multi-vehicle Routing

Emilio Frazzoli[1] and Marco Pavone[2]
[1]Massachusetts Institute of Technology, Cambridge, MA, USA
[2]Stanford University, Stanford, CA, USA

Abstract

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles

moving in a metric space, in order to complete spatially localized, exogenously generated tasks in an efficient way. Control policies depend on several factors, including the definition of the tasks, of the task generation process, of the vehicle dynamics and constraints, of the information available to the vehicles, and of the performance objective. Ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern. Typical performance objectives are represented by measures of quality of service, such as the average or worst-case time a task spends in the system before being completed or the percentage of tasks that are completed before certain deadlines. The scalability of the control policies to large groups of vehicles often drives the choice of the information structure, requiring distributed computation.

Keywords

Cooperative control; Decentralized control; Dynamic routing; Networked robots; Task allocation

Introduction

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles moving in a metric space, in order to complete spatially localized, exogenously generated tasks in an efficient way. Key features of the problem are that tasks arrive *sequentially* over time and planning algorithms should provide *control policies* (in contrast to preplanned routes) that prescribe how the routes should be updated as a function of those inputs that change in real time. This problem is usually referred to as dynamic vehicle routing (DVR). In DVR problems, ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern.

Motivation and Background

As a motivating example, consider the following scenario: a team of unmanned aerial vehicles

(UAVs) is responsible for investigating possible threats over a region of interest. As possible threats are detected, by intelligence, high-altitude or orbiting platforms, or by ground sensor networks, one of the UAVs must visit its location and investigate the cause of the alarm, in order to enable an appropriate response if necessary. Performing this task may require the UAV not only to fly to the possible threat's location but also to spend additional time on site. The objective is to minimize the average time between the appearance of a possible threat and the time one of the UAVs completes the close-range inspection task. Variations may include priority levels, time windows during which the inspection task must be completed, and sensors with limited range.

In order to perform the required mission, the UAVs (or, more in general, mission control) need to repeatedly solve three *coupled* decision-making problems:

1. **Task allocation:** which UAV shall pursue each task? What policy is used to assign tasks to UAVs? How often should the assignment be revised?
2. **Service scheduling:** given the list of tasks to be pursued, what is the most efficient ordering of these tasks?
3. **Loitering paths:** what should UAVs without pending assignments do?

The optimization process must take into account, for example, algebraic or differential constraints (such as obstacle avoidance or bounded curvature, respectively), sensing constraints, communication constraints, and energy constraints. Furthermore, one might require a decentralized control architecture.

DVR problems, including the above UAV routing problem, are generally *intractable* due to their multifaceted combinatorial, differential, and stochastic nature, and consequently solution approaches have been devised that look either at heuristic algorithms or at approximation algorithms with some guarantee on their performance.

Related Problems

DVR problems represent the dynamic counterpart of the well-known static vehicle routing

problem (VRP), whereby (i) a team of n vehicles is required to service a set of n_T "static" tasks in a metric space, (ii) each task requires a certain amount of on-site service, (iii) and the goal is to compute a set of routes that minimizes the cost of servicing the tasks; see Toth and Vigo (2001) for a thorough introduction to this problem. The VRP is *static* in the sense that vehicle routes are computed assuming that no new tasks arrive. The VRP is an important research topic in the operations research community.

Approaches for Multi-vehicle Routing

Broadly speaking, there are three main approaches available in the literature to tackle dynamic vehicle routing problems. The first approach relies on heuristic algorithms. In the second approach, called "competitive analysis approach," routing policies are designed to minimize the *worst-case* ratio between their performance and the performance of an optimal off-line algorithm which has a priori knowledge of the entire input sequence. In the third approach, the routing problem is embedded within the framework of queueing theory. Routing policies are then designed to *stabilize* the system in terms of uniform boundedness of the number of outstanding tasks and to minimize typical queueing-theoretical cost functions such as the *expected time* the tasks remain in the queue. Since the generation of tasks and motion of the vehicles is within an Euclidean space, one can refer to this third approach as "spatial queueing theory."

Heuristic Approach

The main aspect of the heuristic approach is that routing algorithms are evaluated primarily via numerical, statistical and experimental studies, and formal performance guarantees are not available. A naïve, yet reasonable approach to design a heuristic algorithm for DVR would be to adapt classic queueing policies. However, perhaps surprisingly, this adaptation is not at all straightforward. For example, routing algorithms based on a first-come-first-served policy, whereby tasks are fulfilled in the order in which they arrive, are unable to stabilize the system for all stabilizable task arrival rates, in the sense that with such routing algorithms the average number of tasks grows over time without bound, even though there exist alternative routing algorithms that would maintain the number of tasks uniformly bounded (Bertsimas and van Ryzin 1991).

The most widely applied approach is to combine static routing methods (e.g., VRP-like methods, nearest neighbor strategies, or genetic algorithms) and sequential re-optimization, where the re-optimization horizon is chosen heuristically. In particular, greedy nearest neighbor strategies, whose formal characterization still represents an open problem, are known to perform particularly well in some notable cases (Bertsimas and van Ryzin 1991). However, the joint selection of a static routing method and of the re-optimization horizon in the presence of vehicle and task constraints (e.g., differential motion constraints, or task priorities) makes the application of this approach far from trivial. For example, one can show that an erroneous selection of the re-optimization horizon can lead to pathological scenarios where no task *ever* receives service (Pavone 2010). Additionally, performance criteria in dynamic settings commonly differ from those of the corresponding static problems. For example, in a dynamic setting, the time needed to complete a task may be a more important factor than the total vehicle travel cost.

Competitive Analysis Approach

The distinctive feature of the competitive analysis approach is the method used to evaluate an algorithm's performance, which is called *competitive analysis*. In competitive analysis, the performance of a (causal) algorithm is compared to the performance of a corresponding off-line algorithm (i.e., a non-causal algorithm that has a priori knowledge of the entire input) in the worst-case scenario. Specifically, an algorithm is c-competitive if its cost on *any* problem instance is at most c times the cost of an optimal off-line algorithm:

$$\text{Cost}_{\text{causal}}(I) \leq c\, \text{Cost}_{\text{optimal off-line}}(I),$$

$$\textit{for all} \text{ problem instances } I.$$

In the recent past, several dynamic vehicle routing problems have been successfully studied in this framework, under the name of the online traveling repairman problem (Jaillet and Wagner 2006), and many interesting insights have been obtained. However, the competitive analysis approach has some potential disadvantages. First, competitive analysis is a *worst-case* analysis; hence, the results are often overly pessimistic for normal problem instances, and potential statistical information about the problem (e.g., knowledge of the spatial distribution of future tasks) is often neglected. Second, the worst-case analysis usually requires a *finite* horizon problem formulation, which precludes the study of useful properties such as stability. Third, competitive analysis is used to bound the performance relative to an optimal *off-line* algorithm, which, by being non-causal, does *not* belong to the feasible set of routing algorithms one is optimizing over. Hence, with this approach one minimizes the "cost of causality" in the worst-case scenario, but not necessarily the worst-case cost (which would require comparison with an optimal *causal* routing algorithm). Finally, many important real-world constraints for DVR, such as time windows, priorities, differential constraints on vehicle's motion, and the requirement of teams to fulfill a task, have so far proved to be too complex to be considered in the competitive analysis framework (Golden et al. 2008, page 206). Some of these drawbacks have been recently addressed by Van Hentenryck et al. (2009) where a combined stochastic and competitive analysis approach is proposed for a general class of combinatorial optimization problems and is analyzed under some technical assumptions.

Spatial Queueing Theory

Spatial queueing theory embeds the dynamic vehicle routing problem within the framework of queueing theory. Spatial queueing theory consists of three main steps, namely, development of a spatial queueing model, establishment of fundamental limitations of performance, and design of algorithms with performance guarantees. More specifically, the formulation of a model entails detailing four main aspects:

1. A model for the *dynamic* component of the environment: this is usually achieved by assuming that new events are generated (either adversarially or stochastically) by an exogenous process.
2. A model for targets/tasks: tasks are usually modeled as points in a physical environment distributed according to some (possibly unknown) distribution, might require a certain level of on-site service time, and can be subject to a variety of constraints, e.g., time windows, priorities, etc.
3. A model for the vehicles and their motion: besides their number, one needs to specify whether the vehicles are subject to algebraic (e.g., obstacles) or differential (e.g., minimum turning radius) constraints, sensing constraints, and fuel constraints. Also, the control could be centralized (i.e., coordinated by a central station) or decentralized and subject to communication constraints.
4. Performance criterion: examples include the minimization of the waiting time before service, loss probabilities, expectation-variance analysis, etc.

Once the model is formulated, one seeks to characterize fundamental limitations of performance (in the form of lower bounds for the best achievable cost); the purpose of this step is essentially twofold: it allows the quantification of the degree of optimality of a routing algorithm and provides structural insights into the problem. As for the last step, the design of a routing algorithm usually relies on a careful combination of static routing methods with sequential re-optimization. Desirable properties for the static methods are the following: (i) the static problem can be solved (at least approximately) in polynomial time and (ii) the static method is amenable to a statistical characterization (this is essential for the computation of performance bounds). Formal performance guarantees on a routing algorithm are then obtained by

quantifying the ratio between an upper bound on the cost delivered by that algorithm and a lower bound for the best achievable cost. Such a ratio, being an estimate of the degree of optimality of the algorithm, should be close to one and possibly independent of system parameters. The proposed algorithms are finally evaluated via numerical, statistical and experimental studies, including Monte-Carlo comparisons with alternative approaches.

An interesting feature of this approach is that the performance analysis usually yields scaling laws for the quality of service in terms of model data, which can be used as useful guidelines to select system parameters when feasible (e.g., number of vehicles).

In order to make the model tractable, the arrival process of tasks is assumed stationary (with possibly unknown parameters) with statistically independent arrival times. These assumptions, however, can be unrealistic in some scenarios, in which case the competitive analysis approach may represent a better alternative. From a technical standpoint, one should note that spatial queueing models are *inherently* different from traditional, nonspatial queueing models. The main reason is that in spatial queueing models, the "service time" per task has both a *travel* and an *on-site* component. Although the on-site service requirements can often be modeled as "statistically" independent, the travel times are *inherently* statistically coupled. Hence, in contrast to standard queueing models, service times in spatial queueing models are statistically *dependent*, and this deeply affects the solution to the problem.

Pioneering work in this context is that of Bertsimas and van Ryzin (1991), who introduced queueing methods to solve the baseline DVR problem (a vehicle moves along straight lines and visits tasks whose time of arrival, location, and on-site service are stochastic; information about task location is communicated to the vehicle upon task arrival). Next section provides an overview of the application of spatial queueing theory to such simplified DVR problem, referred to in the literature as dynamic traveling repairman problem (DTRP).

Applying Spatial Queueing Theory to DVR Problems

Spatial Queueing Theory Workflow for DTRP

Model

The DTRP, which, incidentally, captures well the salient features of the UAV scenario outlined in the Motivation Section, can be modeled as follows. In a geographical region $\mathcal{Q}$ of area $\mathcal{A}$, a dynamic process generates spatially localized tasks. The process generating tasks is modeled as a spatio-temporal Poisson process, i.e., (i) the time between consecutive generation instants has an exponential distribution with intensity $\lambda > 0$ and (ii) upon arrival, the locations of tasks are independently and uniformly distributed in $\mathcal{Q}$. The location of the new tasks is assumed to be immediately available to a team of n servicing vehicles. The vehicles provide service in $\mathcal{Q}$, traveling at a speed at most equal to v; the vehicles are assumed to have unlimited fuel and task-servicing capabilities. Each task requires an independent and identically distributed amount of on-site service with finite mean duration $\bar{s} > 0$. A task is completed when one of the vehicles moves to its location and performs its on-site service. The objective is to design a *routing policy* that maximizes the quality of service delivered by the vehicles in terms of the average steady-state time delay $\overline{T}$ between the generation of a task and the time it is completed (in general, in a dynamic setting, the focus is on the quality of service as perceived by the "end user," rather than, for example, fuel economies achieved by the vehicles). Other quantities of interest are the average number $\overline{n_{\mathrm{T}}}$ of tasks waiting to be completed and the waiting time $\overline{W}$ of a task before its location is reached by a vehicle. These quantities, however, are related according to $\overline{T} = \overline{W} + \bar{s}$ (by definition) and by Little's law, stating that $\overline{n_{\mathrm{T}}} = \lambda \overline{W}$, for stable queues.

The system is considered stable if the expected number of waiting tasks is uniformly bounded at all times, or equivalently, that tasks are removed from the system at least at the same rate at which they are generated. In the case at hand,

M

the time to complete a task is the sum of the time to reach its location (which depends on the routing policy) plus the time spent at that location in on-site service (which is independent of the routing policy). Since, by definition, the service time is no shorter than the on-site service time $\overline{s}$, then a weaker necessary condition for stability is $\varrho := \lambda \overline{s}/n < 1$; the quantity ϱ measures the fraction of time the vehicles are performing on-site service. Remarkably, it turns out that this is also a sufficient condition for stability, in the sense that, if this condition is satisfied, one can find a stabilizing policy. Note that this stability condition is independent of the size and shape of $\mathcal{Q}$ and of the speed of the vehicles.

Fundamental Limitations of Performance

To derive lower bounds, the main difficulty consists in bounding (possibly in a statistical sense) the amount of time spent to reach a target location. The derivation of these bounds becomes simpler in asymptotic regimes, i.e., looking at cases when $\varrho \to 0^+$ and $\varrho \to 1^-$, which are often called "light-load" and "heavy-load" conditions, respectively.

Consider first the case in which $\varrho \to 0^+$ (light-load regime). A set of n points is called the n-median of $\mathcal{Q}$ if it globally minimizes the expected distance between a random point sampled uniformly from $\mathcal{Q}$ and the closest point in such set. In other words, the n-median of $\mathcal{Q}$ globally minimizes the function

$$
\begin{aligned}
&H_n(p_1, p_2, \ldots, p_n) \\
&\quad := \mathbb{E}\left[\min_{k \in \{1, \ldots, n\}} \|p_k - q\|\right] \\
&\quad = \frac{1}{A} \int_{\mathcal{Q}} \min_{k \in \{1, \ldots, n\}} \|p_k - q\| \, dq.
\end{aligned}
$$

Let H_n^* be the global minimum of this function. Geometric considerations show that H_n^* scales proportionally to $\sqrt{A/n}$.

Incidentally, the n-median of $\mathcal{Q}$ induces a Voronoi partition that is called *Median Voronoi Tessellation*, whose importance will become clear in the next section. Recall that the Voronoi diagram of $\mathcal{Q}$ induced by points $(p_1, \ldots, p_n)$ is defined by

$$
\begin{aligned}
V_i = \Big\{ q \in \mathcal{Q} \big| \, \|q - p_i\| &\le \|q - p_j\|, \; \forall j \ne i, \\
j &\in \{1, \ldots, n\} \Big\},
\end{aligned}
$$

where V_i is the region associated with the i-th "generator" point p_i (see also ▶ Optimal Deployment and Spatial Coverage). The distance H_n^* certainly provides a lower bound on the expected distance traveled by a vehicle to reach a task, and hence one obtains the lower bound

$$
\overline{T} \ge \frac{H_n^*}{v} + \overline{s}.
$$

This lower bound is tight in light-load conditions ($\varrho \to 0^+$), as it will be seen in the next section.

Consider now the case in which $\varrho \to 1^-$ (heavy load). Let $\overline{D}$ be the average travel distance per task for some routing policy. By using arguments from geometrical probability (independent of algorithms), one can show that $\overline{D} \ge \beta_2 \sqrt{A}/\sqrt{2\overline{n_T}}$ as $\varrho \to 1^-$, where β_2 is a constant that will be specified later. As discussed, for stability, one needs $\overline{s} + \overline{D}/v < n/\lambda$. Combining the stability condition with the bound on the average travel distance per task, one obtains

$$
\overline{s} + \frac{\beta_2 \sqrt{A}}{v \sqrt{2\overline{n_T}}} \le \frac{n}{\lambda}.
$$

Since, by Little's law, $\overline{n_T} = \lambda \overline{W}$ and $\overline{T} = \overline{W} + \overline{s}$, one finally obtains (recall that $\varrho = \lambda \overline{s}/n$):

$$
\overline{T} \ge \frac{\beta_2^2}{2} \frac{A}{v^2} \frac{\lambda}{n^2 (1 - \varrho)^2} + \overline{s}, \qquad (\text{as } \varrho \to 1^-).
$$

A salient feature of the above lower bound is that it scales *quadratically* with the number of vehicles (as opposed to the square-root scaling law one has in light-load conditions); note, however, that congestion effects are not included in this model. This bound also shows that the quality of service, which is proportional to $1/(1 - \varrho)^2$, degrades much faster as the target load increases than in nonspatial queueing systems (where the growth rate is proportional to $1/(1 - \varrho)$).

Design of Routing Algorithms

The design of an optimal light-load policy essentially relies on mimicking the proof strategy employed for the light-load lower bound. Specifically, a routing policy whereby (1) one vehicle is assigned to each of the n median locations of $\mathcal{Q}$, (2) new tasks are assigned to the nearest median location and its corresponding vehicle, and (3) each vehicle services tasks according to a first-come-first-served policy is asymptotically optimal, i.e.,

$$\overline{T} \to \frac{H_n}{v} + \overline{s}, \qquad (\text{as } \varrho \to 0^+).$$

Note that under this strategy "regions of dominance" are implicitly assigned to vehicles according to a Median Voronoi Tessellation.

The heavy-load case is more challenging. Consider, first, the following single-vehicle routing policy, based on a partition of $\mathcal{Q}$ into $p \geq 1$ subregions $\{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_p\}$ of equal area A/p. Such a partition can be obtained, e.g., as sectors centered at the median of $\mathcal{Q}$. Define a cyclic ordering for the subregion, such that, e.g., if the vehicle is in region $\mathcal{Q}_i$, the "next" region is $\mathcal{Q}_j$, where j follows i in the cyclic ordering (in other words, $j = (i + 1) \bmod p$).

---

1. If there are no outstanding tasks, move to the median of the region $\mathcal{Q}$.
2. Otherwise, visit the "next" subregion; subregions with no tasks are skipped. Compute a minimum-length path from the vehicle's current position through all the outstanding tasks in that subregion. Complete all tasks on this path, ignoring new tasks generated in the meantime. Repeat.

The problem of computing the shortest path through a number of points is related to the well-known traveling salesman problem (TSP). While the TSP is a prototypically hard combinatorial optimization problem, it is well known that the Euclidean version of the problem can be approximated efficiently (Vazirani 2001). Furthermore, the length $\text{ETSP}(n_T)$ of a Euclidean TSP through

n_T points independently and uniformly sampled in $\mathcal{Q}$ is known to satisfy the following property:

$$\lim_{n_T \to \infty} \text{ETSP}(n_T)/\sqrt{n_T} = \beta_2 \cdot \sqrt{A}, \quad \text{almost surely,}$$

where $\beta_2 \approx 0.712$ is a constant (the same β_2 constant that appeared in the previous section) (Steele 1990).

It can be shown that, using the above routing policy, the average system time $\overline{T}$ satisfies

$$\overline{T} \leq \gamma(p) \frac{A}{v^2} \frac{\lambda}{(1 - \varrho)^2} + \overline{s}, \qquad (\text{as } \varrho \to 1^-),$$

where $\gamma(1) = \beta_2^2$ and $\gamma(p) \to \beta_2^2/2$ for large p. These results critically exploit the statistical characterization of the length of an optimal TSP tour. Hence, the proposed policy achieves a quality of service that is arbitrarily close to the optimal one, in the asymptotic regime of heavy load (and, indeed, also of light load).

The above single-vehicle routing policies can be fairly easily lifted to an efficient multi-vehicle routing policy. The key idea (akin to the one in the light-load case) is to (1) partition the workspace into n *regions of dominance* (with disjoint interiors and whose union is $\mathcal{Q}$), (2) assign one vehicle to each region, and (3) have each vehicle follow a single-vehicle routing policy within its own region. This approach leads to the following multi-vehicle routing policy for the DTRP problem:

---

1. Partition $\mathcal{Q}$ into n regions of dominance of *equal area* and assign one vehicle to each region.
2. Each vehicle executes a single-vehicle DTRP policy in its own subregion.

Using as single-vehicle policy the routing policy described above, the average system time $\overline{T}$ in heavy-load satisfies

$$\overline{T} \leq \gamma(p) \frac{A}{v^2} \frac{\lambda}{n^2 (1 - \varrho)^2} + \overline{s}, \qquad (\varrho \to 1^-).$$

Hence, by comparing this result with the corresponding lower bound, one concludes that a

simple partitioning strategy leads to a multi-vehicle routing policy whose performance is arbitrarily close to the optimal one in heavy load.

Mode of Implementation

The scalability of the control policies to large groups of vehicles often requires a distributed implementation of multi-vehicle routing strategies. For the DTRP, a distributed implementation can be obtained by devising *decentralized algorithms for environment partitioning*. In the solution proposed in Pavone (2010), power diagrams are the key geometric concept to obtain, in a decentralized fashion, partitions suitable for both the light-load case (requiring, as seen before, a Median Voronoi Tessellation) and the heavy-load case (requiring an equal-area partition). The power diagram of $\mathcal{Q}$ is defined as

$$V_i = \left\{ q \in \mathcal{Q} \mid \|q - p_i\|^2 - w_i \leq \|q - p_j\|^2 - w_j, \\ \forall j \neq i, \; j \in \{1, \ldots, n\} \right\},$$

where $(p_i, w_i) \in \mathcal{Q} \times \mathbb{R}$ are a set of "power points" and V_i is the subregion associated with the i-th power point. Note that power diagrams are a generalization of Voronoi diagrams: when all weights are equal, the power diagram and the Voronoi diagram are identical. The basic idea, then, is to associate to each vehicle i a *virtual* power point, which is an artificial (or logical) variable whose value is locally controlled by the i-th vehicle. The cell V_i becomes the region of dominance for vehicle i, and each vehicle updates its own power point according to a *decentralized* gradient-descent law with respect to a coverage function (▶ Optimal Deployment and Spatial Coverage), until the desired partition is achieved. The reader is referred to Pavone (2010) for more details.

Extensions and Discussion

By integrating additional ideas from dynamics, teaming, and distributed algorithms, the spatial queueing theory approach has been recently applied to scenarios with complex models for the tasks such as time constraints, service priori-

ties, translating tasks, and adversarial generation; has been extended to address aspects concerning robotic implementation such as complex vehicle dynamics, limited sensing range, and team forming; and has even been tailored to integrate humans in the design space; see Bullo et al. (2011) and references therein. Despite the significant modeling differences, the "workflow" is essentially the same as in the DTRP: a queueing model that captures the salient features of the problem at hand, characterization of the fundamental limitations of performance, and design of algorithms with provable performance bounds. The last step, as for the DTRP, often involves lifting a single-vehicle policy to a multi-vehicle policy through the strategy of environment partitioning. Within this context, a number of partitioning schemes and corresponding *decentralized* partitioning algorithms relevant to a large variety of DVR problems are discussed in Pavone et al. (2009).

This workflow efficiently and transparently *decouples* the three decision-making problems mentioned in the Introduction Section, i.e., "task allocation," "service scheduling," and "loitering paths." In fact, task allocation is addressed via the strategy of environment partitioning, service scheduling is addressed by applying a single-vehicle routing policy within the individual regions of dominance, and the loitering paths resolve in placing the vehicles at or around specific points within the dominance regions (e.g., the median). Note, however, that in some important cases, e.g., DVR problems where goods have to be transported from a pickup location to a delivery location or where vehicles are differentially constrained and operate in a "congested" workspace, multi-vehicle policies that rely on static partitions perform poorly or are not even feasible (Pavone et al. 2009), and task allocation and service scheduling need to be addressed as tightly coupled.

Through spatial queueing theory one is usually able to characterize the performance of multi-vehicle routing policies in asymptotic regimes. To ensure "satisfactory" performance under general operation conditions, a common strategy is to consider heuristic modifications

to a baseline asymptotically efficient routing policy in such a way that, on the one hand, asymptotic performance is preserved, and, on the other hand, light- and heavy-load performances are "smoothly" and efficiently blended in the intermediate load case. The interested reader can find more information in Bullo et al. (2011).

Summary and Future Directions

The three main approaches available to tackle DVR problems are (i) heuristic algorithms, (ii) competitive analysis, and (iii) spatial queueing theory. Broadly speaking, the competitive analysis approach is well suited when worst-case guarantees are sought, e.g., because there is not enough statistical information about the problem at hand. Spatial queueing theory represents a powerful alternative in cases where it is possible to leverage statistical information and one seeks average-case guarantees. Finally, for some problems the complexity of the model makes an analytical treatment very difficult, in which case the only option is to resort to an heuristic approach (possibly relying on insights derived by applying competitive analysis and/or spatial queueing theory to a simplified version of the problem).

Future directions include the extension of the three aforementioned approaches to increasingly complex problem setups, for example, higher-fidelity vehicle dynamics and environments and sophisticated sensing and communication constraints, novel applications (e.g., search and rescue missions, map maintenance, and pursuit-evasion), and inclusion of game-theoretical tools to address adversarial scenarios. Specifically, for the spatial queueing theory approach, key future directions include the problem of addressing optimality of performance in intermediate regimes (current optimality results are only available either in the light or heavy-load regimes), online estimation of the statistical parameters (e.g., spatial distribution of the tasks), and formulations that take into account second-order moments and large-deviation probabilities.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Flocking in Networked Systems
▶ Networked Systems
▶ Optimal Deployment and Spatial Coverage
▶ Particle Filters

Bibliography

Bertsimas DJ, van Ryzin GJ (1991) A stochastic and dynamic vehicle routing problem in the Euclidean plane. Oper Res 39:601–615

Bullo F, Frazzoli E, Pavone M, Savla K, Smith SL (2011) Dynamic vehicle routing for robotic systems. Proc IEEE 99(9):1482–1504

Golden B, Raghavan S, Wasil E (2008) The vehicle routing problem: latest advances and new challenges. Volume 43 of operations research/computer science interfaces. Springer, New York

Jaillet P, Wagner MR (2006) Online routing problems: value of advanced information and improved competitive ratios. Transp Sci 40(2):200–210

Pavone M (2010) Dynamic vehicle routing for robotic networks. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

Pavone M, Savla K, Frazzoli E (2009) Sharing the load. IEEE Robot Autom Mag 16(2):52–61

Steele JM (1990) Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space. Math Oper Res 15(4):749

Toth P, Vigo D (eds) (2001) The vehicle routing problem. Monographs on discrete mathematics and applications. SIAM, Philadelphia. ISBN:0898715792

Van Hentenryck P, Bent R, Upfal E (2009) Online stochastic optimization under time constraints. Ann Oper Res 177(1):151–183

Vazirani V (2001) Approximation algorithms. Springer, New York

M

N

Nash equilibrium

▸ Strategic Form Games and Nash Equilibrium

Network Games

R. Srikant
Department of Electrical and Computer
Engineering and the Coordinated Science Lab,
University of Illinois at Urbana-Champaign,
Champaign, IL, USA

Abstract

Game theory plays a central role in studying systems with a number of interacting players competing for a common resource. A communication network serves as a prototypical example of such a system, where the common resource is the network, consisting of nodes and links with limited capacities, and the players are the computers, web servers, and other end hosts who want to transfer information over the shared network. In this entry, we present several examples of game-theoretic interaction in communication networks and present a simple mathematical model to study one such instance, namely, resource allocation in the Internet.

Keywords and Phrases

Introduction

A communication network can be viewed as a collection of resources shared by a set of competing users. If the network were totally unregulated, then each user would attempt to grab as many resources in the network as possible, resulting in poor network performance, a situation commonly referred to as the *tragedy of the commons* (Hardin 1968). In reality, there is a carefully designed set of network protocols and pricing mechanisms which provide incentives to users to act in a socially responsible manner. Since game theory is the mathematical discipline which studies the interactions between selfish users, it is a natural tool to use to design these network control mechanisms. We now provide a few examples of network problems which naturally lend themselves to game-theoretic analysis. Later, we will elaborate on the game-theoretic formulation of one of these examples.

- *Resource Allocation:* A network such as the Internet is a collection of links, where each link has a limited data-carrying capacity, usually measured in bits per second.

The Internet is shared by billions of users, and the actions of these users have to be regulated so that they share the resources in the network in a fair manner. Equivalently, this problem can be viewed as one in which a network designer has to design a collection of protocols so that the users of the network can equitably allocate the available resources among themselves without the intervention of a central authority. Such protocols are built into every computer connected to the Internet today, to allow for seamless operation of the network. The problem of designing such protocols can be posed as a game-theoretic problem in which the players are the network and the traffic sources using the network (Kelly 1997).

- *Routing Games:* Finding appropriate routes for each user's data traffic is a particular form of resource allocation mentioned above. However, routing has applications beyond communication networks (with the other major application area being transportation networks), so it is useful to discuss routing separately. In communication networks, each user may attempt to find the minimum-delay route for its traffic, with help from the network, to minimize the delay experienced by its packets. In a transportation network, each automobile on the road attempts to take the path of least congestion through the network. An active area of research in game theory is one which tries to understand the impact of individual user decisions on the global performance of the network (Roughgarden 2005). An interesting result in this regard is the *Braess paradox* which is an example of a road transportation network in which the addition of a road leads to increased delays when each user selfishly choose a route to minimize its delay. Of course, if routes are chosen to minimize the overall delay experienced in the network such a paradox will not arise.

- *Peer-to-Peer Applications:* Many studies have indicated that file sharing between users (also known as peers) directly, without using a centralized web site such as YouTube, is a dominant source of traffic in the Internet.

For such a peer-to-peer service to work, each peer should not only download files from others, but should also be willing to sacrifice some of its resources to upload files to others. Naturally, peers would prefer to only download and not upload to minimize their resource usage. The design of incentive schemes to induce users to both download and upload files is another example of a game-theoretic problem in a network (Qiu and Srikant 2004).

- *Network Economics:* In addition to end-user interaction, Internet service providers (ISPs) have to interact with each other to allow their customers access to all the web sites in the world. For example, one ISP may have a customer who wants to access a web site connected to another ISP. In this case, the data traffic must cross ISP boundaries, and thus, one ISP has to transport data destined for a customer of another ISP. Thus, ISPs must be willing to contribute resources to satisfy the needs of customers who do not directly pay them. In such a situation, ISPs must have bilateral agreements (commonly known as *peering agreements*) to ensure that the selfish interest of each ISP to minimize its resource usage is aligned with the needs of its customers. Again, game theory is the right tool to study such inter-ISP interactions (Courcoubetis and Weber 2003).

- *Spectrum Sharing:* Large portions of the radio spectrum are severely underutilized. Typically, portions of the spectrum are assigned to a primary user, but the primary user does not use it most of the time. There has been a surge of interest recently in the concept of *cognitive radio,* whereby radios are cognitive of the presence or absence of the primary user, and when the primary user is absent, another radio can use the spectrum to transmit its data. When there are many users and the available spectrum is split into many channels, it is impossible for users to perfectly coordinate their transmissions to achieve maximum network utilization. In these situations, game-theoretic protocols which take into account the noncooperative behavior of the users can be

designed to allow secondary users to access the available channels as efficiently as possible (Saad et al. 2009).

In the next section, we will elaborate on one of the applications above, namely, resource allocation in the Internet, and show how game-theoretic modeling can be used to design fair resource sharing.

Resource Allocation and Game Theory

Consider a network consisting of L links, with link l having capacity c_l. Suppose that there are R users sharing the network, with each user r being characterized by a set of links which connect the user's source to its destination. Since each user uses a fixed route in our model, we will use r to denote both the user and the route used by the user. We use the notation $l \in r$ to denote that link l is a part of route r. Let x_r denote the rate at which user r transmits data. Thus, we have the following natural constraints, which state that the total data rate on any link must be less than or equal to the capacity of the link:

$$\sum_{r:l \in r} x_r \le c_l, \qquad \forall l. \tag{1}$$

Associated with each user is a concave utility function $U_r(x_r)$ which is the utility that user r derives by transmitting data at rate x_r. The network utility maximization problem is to solve

$$\max_{x \ge 0} \sum_r U_r(x_r), \tag{2}$$

subject to the constraint (1). In (2), x denotes the vector $(x_1, x_2, \ldots, x_R)$ and $x \ge 0$ means that each component of x must be greater than or equal to zero. Note that the goal of the network in (2) is to maximize the sum of the utilities of the users in the network.

Let p_l be the Lagrange multiplier corresponding to the capacity constraint in (1) for link l. Then the Lagrangian for the problem is given by

$$L(x, p) = \sum_r U_r(x_r) - \sum_l p_l(y_l - c_l), \tag{3}$$

where we have used the notation $y_l := \sum_{r:l \in r} x_r$ to denote the total data rate on link l. If p is known, then the optimal x can be calculate by solving

$$\max_{x \ge 0} L(x, p).$$

Notice that the optimal solution for each x_r can be obtained by solving

$$\max_{x_r \ge 0} U_r(x_r) - q_r x_r, \tag{4}$$

where $q_r = \sum_{l \in r} p_l$. Thus, if the Lagrange multipliers are known, then the network utility maximization can be interpreted as a game in the following manner. Suppose that the network charges each user q_r dollars for every bit transmitted by user r though the network. Then, $q_r x_r$ is the dollars per second spent by the user if x_r is measured in bits per second. Interpreting $U_r(x_r)$ as the dollars per second that the user is willing to pay for transmitting at rate x_r, the optimization problem in (4) is the problem faced by user r which wants to maximize its net utility, i.e., utility minus cost. Thus, the individual optimal solution for each user is also the solution to the network utility maximization problem. The above game-theoretic interpretation of the network utility maximization problem is somewhat trivial since, given the p_l's or q_r's, there is no interaction between the users. Of course, this interpretation relies on the ability of the network to compute p. We next present a scheme to compute p, which couples the users closely and thus allows for a richer game-theoretic interpretation.

Suppose that the network wants to compute p but does not have access to the utility functions of the users. The network asks each user r to bid an amount w_r which is interpreted as the dollars per second that the user is willing to pay. The network then assumes that user r's utility function is $w_r \log x_r$ and solves the network utility maximization. While this choice of utility function may seem arbitrary, the resulting solution x has a number of attractive properties, including a form of fairness called *proportional fairness*. The proportionally fair resource allocation solution to (4) is given by

$$\frac{w_r}{x_r} = q_r. \tag{5}$$

The network then allocates rate x_r to user r and charges q_r dollars per bit. From (5), the amount charged to user r per second is w_r, thus satisfying the original interpretation of w_r. Knowing that the network charges users in this manner, how might a user choose its bid w_r? Recall that user r's goal is to solve (4). Substituting from (5), the problem in (4) can be rewritten as

$$\max_{w_r \geq 0} U_r \left(\frac{w_r}{q_r}\right) - w_r. \tag{6}$$

Thus, the users' problem of selecting w can be viewed as a game, with each user's objective given by (6). Note that q_r is given by (5) and thus depends on all the w_r's. Depending upon the application, the game can be solved under one of two assumptions:

• *Price-Taking Users:* Under this assumption, users are assumed to take the price q_r as given, i.e., they do not attempt to infer the impact of their actions on the price. This is a reasonable assumption in a large network such as the Internet, where the impact of a single user on the link prices is negligible, and it is practically impossible for any user to infer the impact of its decisions on the prevailing price of the network resources. When the users are price taking, the socially optimal solution, i.e., the solution to the network utility maximization problem, coincides with the Nash equilibrium of the game. To see this, note that the solution to (6) is given by

$$\frac{1}{q_r} U_r' \left(\frac{w_r}{q_r}\right) - 1 = 0,$$

under the assumption that the utility function is differentiable and the solution is bounded away from zero. Using (5), this equation reduces to

$$U_r'(x_r) = q_r,$$

which maximizes the Lagrangian (3). It is not difficult to see that the complementary slackness equations in the Karush-Kuhn-Tucker conditions are satisfied since the constraints for (2) and the proportionally fair solution are the same. Thus, under the price-taking assumption, the equilibrium of the game solution is the same as the socially optimal solution provided the network computes q_r using the proportionally fair resource allocation formulation.

• *Strategic Users:* In networks where the number of users is small, it may be possible for each user to know the topology of the network, and thus, each user may be able to solve for the proportionally fair resource allocation if it has access to other users' bids. In other words, it may be possible to compute a Nash equilibrium by taking into account the impact of the w_r's on the q_r's. When the users are strategic, the socially optimal solution could be quite different from the Nash equilibrium. The ratio of the network utility under the socially optimal solution to the network utility under a Nash equilibrium is called the *price of anarchy*.

There is a rich literature associated with both interpretations of the network congestion game. In the case of price-taking users, much of the emphasis in the literature has been on designing distributed algorithms to achieve the socially optimal solution (Shakkottai and Srikant 2007). In the case of strategic users, the focus has been on characterizing the price of anarchy (Johari and Tsitsiklis 2004; Yang and Hajek 2007).

Summary and Future Directions

We have presented a number of applications which involve the interactions of selfish users over a network. For the resource allocation application, we have also described how simple mathematical models can be used to provide incentives for users to act in a socially optimal manner. In particular, we have shown that, under the reasonable price-taking assumption and an appropriate computation of link prices, selfish users automatically maximize network utility. In the case where the users are strategic, the goal is to characterize the price of anarchy.

Moving forward, two areas which require considerable further research are the following: (i) inter-ISP routing and (ii) spectrum sharing. The Internet is a fairly reliably network, and any unreliability often arises due to routing issues among ISPs. As mentioned in the introduction, peering arrangements between ISPs are necessary to make sure that ISPs carry each others' traffic and are appropriately compensated for it, either through reciprocal traffic-carrying agreements or actual monetary transfer. Thus, the policy that an ISP uses to route traffic may be governed by these peering agreements. The more complicated these policies are, the more chances there are for routing misconfigurations that lead to service interruptions. This interplay between policies and technology in the form of routing algorithms is an interesting topic for further study.

Cognitive radios and spectrum sharing are expected to be significant technological components of future wireless networks. Designing algorithms for selfish radios to share the available spectrum while respecting the rights of the primary user of the spectrum is a challenge that requires considerable further attention. This area of research requires one to combine sensing technologies to sense the presence of other users with game-theoretic models to ensure fair channel access to the secondary users, subject to the constraint that the primary user should not be affected by the presence of the secondary users.

Cross-References

Bibliography

Courcoubetis C, Weber R (2003) Pricing communication networks: economics, technology and modelling. Wiley, Hoboken
Hardin G (1968) The tragedy of the commons. Science 162:1243–1248
Johari R, Tsitsiklis JN (2004) Efficiency loss in a network resource allocation game. Math Oper Res 29:407–435
Kelly FP (1997) Charging and rate control for elastic traffic. Eur Trans Telecommun 8:33–37
Qiu D, Srikant R (2004) Modeling and performance analysis of BitTorrent-like peer-to-peer networks. Proc ACM SIGCOMM ACM Comput Commun Rev 34:367–378
Roughgarden T (2005) Selfish routing and the price of anarchy. MIT Press, Cambridge
Saad W, Han Z, Debbah M, Hjorungnes A, Basar T (2009) Coalitional game theory for communication networks: a tutorial. IEEE Signal Process Mag 26(5):77–97
Shakkottai S, Srikant R (2007) Network optimization and control. NoW Publishers, Boston-Delft
Yang S, Hajek B (2007) VCG-Kelly mechanisms for allocation of divisible goods: adapting VCG mechanisms to one-dimensional signals. IEEE J Sel Areas Commun 25:1237–1243

Networked Control Systems: Architecture and Stability Issues

Linda Bushnell[1] and Hong Ye[2]
[1]Department of Electrical Engineering, University of Washington, Seattle, WA, USA
[2]The Mathworks, Inc., Natick, MA, USA

Abstract

When shared, band-limited, real-time communication networks are employed in a control system to exchange information between spatially distributed components, such as controllers, actuators, and sensors, it is categorized as a networked control system (NCS). The primary advantages of a NCS are reduced complexity and wiring, reduced design and implementation cost, ease of system maintenance and modification, and efficient data sharing. In addition, this unique architecture creates a way to connect the cyberspace to the physical space for remote operation of systems. The NCS architecture allows for performing more complex tasks, but also requires taking the network effects into account when designing control laws and stability conditions. In this entry, we review significant results on the architecture and stability analysis of a NCS. The results presented address communication network-induced challenges such as time delays, scheduling, and information packet dropouts.

N

Keywords

Architecture; Networked control system; Stability

Introduction

From the washing machine, air conditioner, and microwave oven to the telephone, stereo, and automobile, embedded computers are present in the modern home. In a factory environment, there are thousands of networked smart sensors and actuators with embedded processors, working to complete a coordinated task. The trend in manufacturing plants, homes, buildings, aircraft, and automobiles is toward distributed networking. This trend can be inferred from many proposed or emerging network standards, such as controller area network (CAN) for automotive and industrial automation, BACnet for building automation, PROFIBUS and WorldFIP fieldbus for process control, and IEEE 802.11, and Bluetooth wireless standards for applications such as mobile sensor networks, HVAC systems, and unmanned aerial vehicles.

The traditional dedicated point-to-point wired connection in control systems has been successfully implemented in industry for decades. With the advance of communication network and hardware technologies, it is common to integrate the communication network into the control system to replace the dedicated point-to-point connection to achieve reduced weight and power, lower cost, simpler installation and maintenance, and higher reliability, to name a few advantages. For example, a typical new automobile has two controller area networks (CANs): a high-speed one in front of the firewall for the engine, transmission, and traction control and a low-speed one for locks, windows, and other devices (Johansson et al. 2005).

The conventional definition of a networked control system (NCS) is as follows: When a feedback control system is closed via a communication channel, which may be shared with other nodes outside the control system, then the control system is called a NCS. A NCS can also be described as a feedback control system where the control loops are closed through a real-time communication network.

Architecture of Networked Control Systems

The architecture of a NCS consists of a band-limited, digital communication network physically and electronically integrated with a spatially distributed control system, operated on a given plant. Digital information, such as controller signals, actuator signals, sensor signals, and operator input, is transmitted via the network. The components connected by the network include all nodes of the control system, such as the supervisory (or "network owner") computer, controller software and hardware, actuators, and sensors. In this structure, the feedback control system's loops are closed over the shared communication network.

The communication network can be wired or wireless and may be shared with other unrelated nodes outside the control system. As illustrated in Fig. 1, the shared communication channel, which multiplexes signals from the sensors to the controllers and/or from the controllers to the actuators, serves many other uses besides control. Each of the system components directly connected to the network via a network interface is denoted a physical node. Besides the network interface, the sensors and actuator nodes are typically smart nodes with embedded microprocessors. Sometimes, the controller is colocated with the smart actuator. Several key issues make networked control systems distinct from traditional control systems (Hespanha et al. 2007; Yang 2006).

Band-Limited Channels

Bandwidth limitation of the shared communication channel requires that all nodes in the network must share (e.g., time sharing or frequency sharing, etc.) the common network resource without interfering with each other.

Sampling and Delays

In a NCS, the plant outputs are sampled by the sensors, which can convert continuous-time
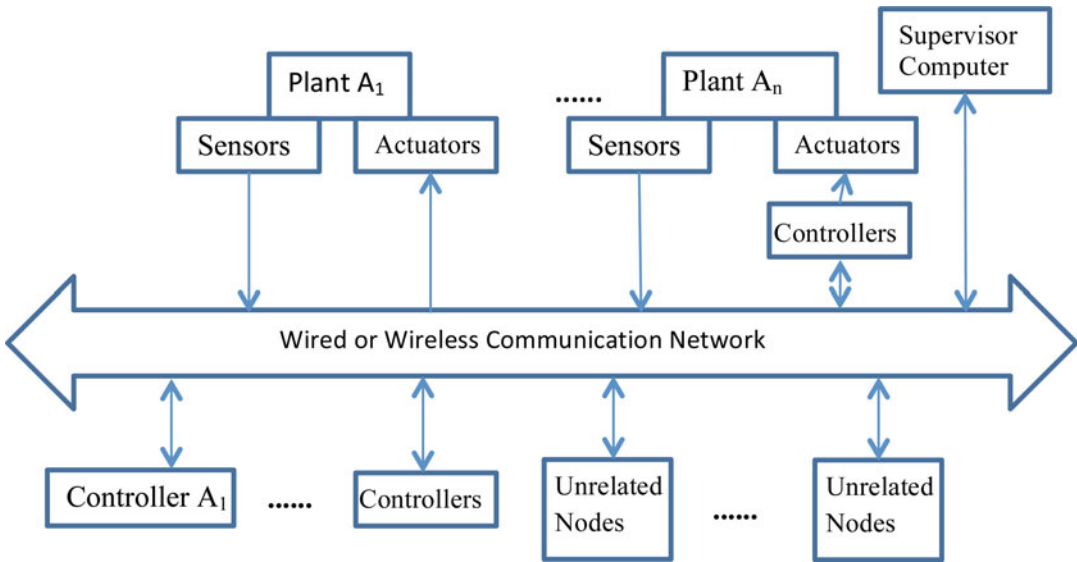
Networked Control Systems: Architecture and Stability Issues, Fig. 1 A general networked control system (NCS) architecture

analog signals to digital signals; perform preprocessing, filtering, and encoding; and package the data signal so that it is ready for transmission. After winning the medium access control and being transmitted over the network, the package containing the sampled data signal arrives at the receiver side, which could be a controller or a smart actuator with a controller collocated with it. The receiver unpacks and decodes the signal. This process is quite different from the traditional periodic sampling in digital control. The overall delay between sampling and the eventual decoding of the transmitted packet at the receiver can be time varying and random due to both the network access delay (i.e., the time it takes for a shared network to accept the data) and the transmission delay (i.e., the time during which data are in transit inside the network). This also depends on the highly variable network conditions, such as congestion and channel quality. In some NCSs, the data transmitted are time stamped, which means that the receiver may have an estimate of the delay's duration and could take appropriate corrective action. Given the rapid advance of embedded computation and communication hardware technology today, the transmission delay in

many embedded systems can be neglected when compared with the magnitude of network access delay.

Packet Dropouts

It is possible in a NCS that a packet may be lost while it is in transit through the network. The packet that contains important sampling data or control signals may drop occasionally due to transmission errors of the physical network link, message collision, or node failures, to name a few. Overflow in queue or buffer can lead to network congestion and package loss. Thus, the use of queues is not favored by NCSs in general. Packet dropouts also happen if the receiver discards outdated arrivals that have long delays. Most network protocols are equipped with transmission-retry mechanisms, such as TCP, that guarantee the eventual delivery of packets. These protocols, unfortunately, are not appropriate for a NCS since the retransmission of old sensor data or calculated control signals is generally not very useful when new, time-critical data are available. Using selected old data for estimation or prediction is an exception, where old data may be packaged with the new

data in one packet. It is advantageous to discard the old, un-transmitted data and transmit a new packet if and when it becomes available. In this way, the controller always receives fresh data for its control calculation, and the actuator always executes the up-to-date command to control the plant.

Modeling Errors, Uncertainties, and Disturbances

In a distributed NCS, modeling errors, uncertainties, and disturbances always exist when using the mathematical model to describe the physical process. These factors may lead to a major impact on the overall system performance and cause failure in fulfilling the desired objectives. Wang and Hovakimyan (2013) proposed a reference model-based architecture to decouple the design of controller and communication schemes. A reference model is introduced in each subsystem as a bridge to build the connection between the real system and an ideal model, free of uncertainties. The closeness between the real system and the reference model is associated only with plant uncertainties, and the difference between the reference model and the ideal model is only in the communication constraints.

Stability of Networked Control Systems

The stability of a control system is often extremely important and is generally a safety requirement. Examples include the control of rockets, robots, airplanes, automobiles, or ships. Instability in any one of these systems can result in an unimaginable accident and loss of life. The stability of a general dynamical system with no input can be described with the Lyapunov stability criteria, which is stated as follows: A linear system is stable if its impulse response approaches zero as time approaches infinity or if every bounded input produces a bounded output.

When sensors, controllers, and actuators are not colocated and use a shared network to communicate, the feedback loop of a NCS is closed over the network. Network-induced,

variable delays, and packet dropouts can degrade the performance of a NCS. For example, the NCS may have a longer settling time or bigger overshoot in the step response. Furthermore, the NCS may become unstable when delays and/or packet dropouts exceed a certain range. Designers choosing to use a NCS architecture, however, are motivated not by performance but by cost, maintenance, and reliability gains.

Band-Limited Channels

Inspired by Shannon's results on the maximum bit rate that a communication channel can carry reliably, a significant research effort has been devoted to the problem of determining the minimum bit rate that is needed to stabilize a system through feedback over a finite capacity channel (Baillieul 1999; Nair and Evans 2000; Tatikonda and Mitter 2004; Wong and Brockett 1999; Baillieul and Antsaklis 2007). This has been solved exactly for linear plants, but only conservative results have been obtained for nonlinear plants. The data-rate theorem that quantifies a fundamental relationship between unstable physical systems and the rate at which information must be processed in order to stably control them was proved independently under a variety of assumptions. Minimum bit rate and quantization becomes especially important for networks designed to carry very small packets with little overhead, because encoding measurements or actuation signals with less bits can save network bandwidth.

Most of the NCS stability results presented here, however, are based on the observation that the channel can transmit a finite number of packets per unit of time (packet rate) and each packet can carry certain number of bits in the data field. The packets on a real-time control network typically are frequent and have small data segments compared to their headers. For example, a CAN II packet with a single 16-bit data sample has fixed 64 bits of overhead associated with identifier, control field, CRC, ACK field, and frame delimiter, resulting in 25 % utilization, and this utilization can never exceed 50 % (data field length is limited to 64 bits). Thus, the quantization effects imposed by the communication networks are generally ignored.

Network-Induced Delays

A significant number of results have attempted to characterize a maximum upper bound on the sampling or transmission interval for which stability of the NCS can be guaranteed. The upper bound is sometimes called the maximum allowable transfer interval (MATI) (Walsh 2001a). These results implicitly attempt to minimize the packet rate or schedule the traffic of the control network that is needed to stabilize a system through feedback. The general approach is to design the controller using established techniques, considering the network to be transparent, and then to analyze the effect of the network on closed-loop system performance and stability.

The NCS with a linear time-invariant (LTI) plant/controller pair and one-channel feedback (see Fig. 2) can be modeled by the following continuous-time system, where x includes the states of the plant and the controller, $x(t) = [x_p(t), x_c(t)]^T$:

$$\dot{x} = Ax + B\hat{y}, \quad y = C(x) \qquad (1)$$

$$\hat{y}(t) = \begin{cases} \hat{y}_{k-1}, t \in [t_k, t_k + \tau_k) \\ \hat{y}_k, t \in [t_k + \tau_k; t_{k+1}) \end{cases} \qquad (2)$$

The signal y is a vector of sensor measurements and $\hat{y}$ is the input to a continuous-time controller collocated with the actuators. Alternatively, $\hat{y}$ can be viewed as the input to the actuators and y as the desired control signal computed by a controller collocated with the sensors. The signal y(t) is sampled at times $\{t_k : k \in N\}$ and the samples $y(k) := y(t_k)$ are sent through the network. But the samples arrive at the destination after a (possibly variable) delay of τ_k, where we assume that the network delays are always smaller than one sampling interval. For periodic sampling and constant delays, a sufficient and necessary condition for exponential stability of the NCS (Eqs. 1 and 2) was derived (Zhang et al. 2001). By using the augmented state space model and based on the stability of nonlinear hybrid systems, they also proved the sufficient condition for stability of the NCS in the time-invariant case.

If we now assume the sampling intervals are constant and the computation and transmission delays are negligible, then the variable network access delays serve as the main source of delays in a NCS (Lin et al. 2003, 2005). Using average dwell time results for discrete switched systems, Zhai et al. (2002) provided conditions such that NCS stability is guaranteed. Also, the authors consider robust disturbance attenuation analysis for this class of NCSs.

When the network delay is not constant or when the signal y(t) is sampled in a nonperiodic fashion, the system (1) and (2) is not time invariant and one needs a Lyapunov-based argument to prove its stability. Zhang and Branicky (2001) derived the sufficient condition to ensure the NCS in Fig. 2 is exponentially stable. They also proposed a randomized algorithm to find the largest value of sampling interval for which stability can be guaranteed.

For a model-based NCS with state and output feedback, an explicit model of the plant is used to produce an estimate of the plant state behavior between transmission times (Montestruque and Antsaklis 2004). Sufficient conditions for Lyapunov stability are derived for a model-based NCS when the controller/actuator is updated with the sensor information at nonconstant time intervals. A NCS with transmission times that are driven by a stochastic process with identically independently distributed and Markov-chain-driven transmission times almost sure stability and mean-square sufficient conditions for stability are introduced. Onat et al. (2011) adapted above stability results to model-based predictive NCSs with realistic structure assumptions.
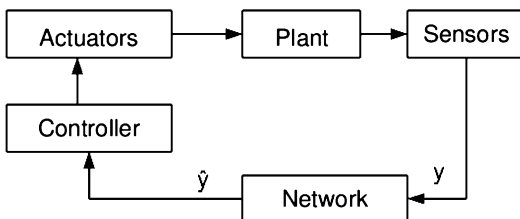
Networked Control Systems: Architecture and Stability Issues, Fig. 2 A NCS architecture with one-channel feedback (controller collocated with actuator)
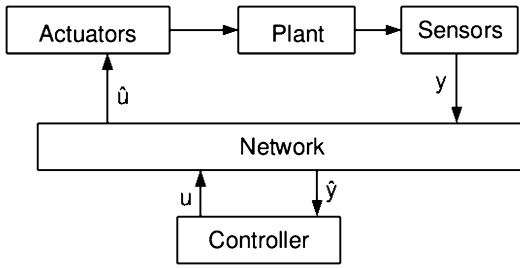
Networked Control Systems: Architecture and Stability Issues, Fig. 3 A NCS architecture with two-channel feedback

Control Network Scheduler

In general a Multi-Input/Multi-Output (MIMO) NCS with two-channel feedback, both the sampled plant output and controller output are transmitted via a network (see Fig. 3). Because of the network, only the reported output y(t) is available to the controller and its prediction processes; similarly, only û(t) is available to the actuators on the plant. We label the network-induced error

$$e(t) := [\hat{y}(t), \hat{u}(t)]^T - [y(t), u(t)]^T$$

and the combined state of controller and plant $x(t) = [x_p(t), x_c(t)]^T$. The state of the entire NCS is given by $z(t) = [x(t), e(t)]^T$. Following this general approach, the controller is designed using established techniques without considering the presence of the network.

The behavior of the network-induced error $e(t)$ is mainly determined by the architecture of the NCS and the scheduling strategy. In the special case of one-package transmission, there is only one node transmitting data on the network; therefore, the entire vector $e(t)$ is set to zero at each transmission time. For multiple nodes transmitting measured outputs $y(t)$ and/or computed inputs $u(t)$, the transmission order of the nodes depends on the scheduling strategy chosen for the NCS. In other words, the scheduling strategy decides which components of $e(t)$ are set to zero at the transmission times.

Static and dynamic schedulers (a.k.a. protocols) are two main categories used in a NCS. When the network resource or transmission order are pre-allocated or determined before run-time,

it is called a static scheduler, such as round-robin scheduling. A dynamic scheduler determines the network allocation while the system runs. A novel dynamic network scheduler, try-once-discard (TOD) and several variations were introduced for wired and wireless NCSs (Walsh and Ye 2001; Ye et al. 2001). For linear and nonlinear NCSs with the new dynamic and commonly used static schedulers, an analytic proof of global exponential stability of a MIMO NCS was provided (Walsh 2001a; Walsh et al. 2001b). Simulation and experiment results showed that the dynamic schedulers outperform static schedulers in terms of NCS performance, e.g., a bigger MATI.

Nesic and Teel (2004a,b) generalize the above results by considering a nonlinear NCS with external disturbances and more general class of protocols (or schedulers). They considered a new class of Lyapunov uniformly globally asymptotically stable (UGAS) protocols in a NCS. It is shown that if the controller is designed without taking into account the network, it yields input-to-state stability (ISS) with respect to external disturbances (not necessarily with respect to the network-induced error), and then the same controller will achieve semi-global practical ISS for the NCS when implemented via the network with a Lyapunov UGAS protocol. Moreover, the ISS gain is preserved. The adjustable parameter with respect to which semi-global practical ISS is achieved is the MATI between transmission times. The authors also studied the input–output L_p stability of a NCS for a large class of network scheduling protocols. It is shown that polling, static protocols, and dynamic protocol such as TOD belong to this class. Results in Nesic and Teel (2004a) provide a unifying framework for generating new scheduling protocols that preserve L_p stability properties of the system, if a design parameter is chosen to be sufficiently small. The most general version of these results can also be used to model a NCS with data packet dropouts. The proof technique used is based on the small gain theorem and lends itself to an easy interpretation.

A framework for analyzing the stability of a general nonlinear NCS with disturbances in

the setting of L_p stability was provided by Tabbara et al. (2007). Their presentation provides sharper results for both gain and MATI than previously obtainable and details the property of uniformly persistently exciting scheduling protocols. This class of protocols was shown to lead to stability for high enough transmission rates. This was a natural property to demand, especially in the design of wireless scheduling protocols. The property is used directly in a novel proof technique based on the notions of vector comparison and (quasi)-monotone systems. Via simulations, analytical, and numerical comparison, it is verified that the uniform persistence of excitation property of protocols is, in some sense, the "finest" property that can be extracted from wireless scheduling protocols.

Delays and Packet Dropouts

Packet dropouts can be modeled as either stochastic or deterministic phenomena. For a one-channel feedback NCS, Zhang and Branicky (2001) consider a deterministic dropouts model, with packet dropouts occurring at an asymptotic rate. Stability conditions were studied for a NCS with deterministic and stochastic dropouts (Seiler and Sengupta 2005).

Sometimes, the NCS was characterized as a continuous-time delayed differential equation (DDE) with the time-varying delay $\tau(t)$. One important advantage is that the equations are still valid even when the delays exceed the sampling interval. Researchers successfully used the Lyapunov–Krasovskii (Yue et al. 2004) and the Razumikhin theorems (Yu et al. 2004) to study the stability of a NCS that is modeled as DDEs.

Summary and Future Directions

This article introduced the concept of a networked control system and its general architecture. Several key issues specific to a NCS, such as band-limited channels, network-induced delays, and information packet dropouts, were explained. The stability condition of a NCS with various network effects was discussed with several common modeling techniques.

In terms of future directions, there has been significant effort in analyzing networked control systems with variable sampling rate, but most results investigate the stability for a given worst-case interval between consecutive sampling times, leading to conservative results. An open area of research would be to look at methods that take into account a stochastic characterization for the inter-sampling times. Substantial work has also been devoted to determining the stability of a NCS, as described in this article. Possible open areas of research would be to consider design issues related to the joint stability and performance of the system. The design and development of controllers for a NCS is also an open area of research. In designing a controller for a NCS, one has to take into account the challenges introduced by the communication network. Only afterward can analysis of the whole system take place.

Cross-References

▶ Data Rate of Nonlinear Control Systems and Feedback Entropy
▶ Information and Communication Complexity of Networked Control Systems
▶ Networked Control Systems: Estimation and Control over Lossy Networks
▶ Networked Systems
▶ Quantized Control and Data Rate Constraints

Bibliography

Baillieul J (1999) Feedback designs for controlling device arrays with communication channel bandwidth constraints. In: Lecture notes of the fourth ARO workshop on smart structures, Penn State University

Baillieul J, Antsaklis PJ (2007) Control and communication challenges in networked control systems. Proc IEEE 95(1):9–28

Hespanha JP, Naghshtabrizi P, Xu Y (2007) A survey of recent results in networked control systems. Proc IEEE 95(1):138–162

Johansson KH, Torngren M, Nielsen L (2005) Vehicle applications of controller area network. In: Levine WS, Hristu-Varsakelis D (eds) Handbook of networked and embedded control systems. Birkhäuser, Boston, pp 741–765

Lin H, Zhai G, Antsaklis PJ (2003) Robust stability and disturbance attenuation analysis of a class of networked control systems. In: 42nd IEEE conference on decision and control, Maui, vol 2, pp 1182–1187

Lin H, Zhai G, Fang L, Antsaklis PJ (2005) Stability and H_1 performance preserving scheduling policy for networked control systems. In: Proceedings 16th IFAC world congress, Prague

Montestruque LA, Antsaklis PJ (2004) Stability of model-based networked control systems with time-varying transmission times. IEEE Trans Autom Control 49(9):1562–1572

Nair GN, Evans RJ (2000) Stabilization with data-rate-limited feedback: tightest attainable bounds. Syst Control Lett 41(1):49–56. Elsevier

Nesic D, Teel A (2004a) Input-output stability properties of networked control systems. IEEE Trans Autom Control 49(10):1650–1667

Nesic D, Teel A (2004b) Input-to-state stability of networked control systems. Automatica 40(12):2121–2128

Onat A, Naskali T, Parlakay E, Mutluer O (2011) Control over imperfect networks: model-based predictive networked control systems. IEEE Trans Ind Electron 58:905–913

Seiler P, Sengupta R (2005) An H_∞ approach to networked control, IEEE Trans Autom Control 50(3):356–364

Tabbara M, Nesic D, Teel A (2007) Stability of wireless and wireline networked control systems. IEEE Trans Autom Control 52(9):1615–1630

Tatikonda S, Mitter S (2004) Control under communication constraints. IEEE Trans Autom Control 49(7):1056–1068

Walsh G, Ye H (2001) Scheduling of networked control systems. IEEE Control Syst Mag 21(1): 57–65

Walsh G, Ye H, Bushnell L (2001a) Stability analysis of networked control systems. IEEE Trans Control Syst Technol 10(3):438–446

Walsh G, Beldiman O, Bushnell L (2001b) Asymptotic behavior of nonlinear networked control systems. IEEE Trans Autom Control 44:1093–1097

Wang X, Hovakimyan N (2013) Distributed control of uncertain networked systems: a decoupled design. IEEE Trans Autom Control 58(10):2536–2549

Wong WS, Brockett RW (1999) System with finite communication bandwidth constraints-II: stabilization with limited information feedback. IEEE Trans Autom Control 44(5):1049–1053

Yang TC (2006) Networked control system: a brief survey. IEE Proc Control Theory Appl 153(4):403–412

Ye H, Walsh G, Bushnell L (2001) Real-time mixed-traffic wireless networks. IEEE Trans Ind Electron 48(5):883–890

Yu M, Wang L, Chu T, Hao F (2004) An LMI approach to networked control systems with data packet dropout and transmission delays. 43rd IEEE conference on decision and control, Paridise Island, Bahamas, vol 4, pp 3545–3550

Yue D, Han QL, Peng C (2004) State feedback controller design for networked control systems. IEEE Trans Circuits Syst 51(11):640–644

Zhai G, Hu B, Yasuda K, Michel A (2002), Qualitative analysis of discrete-time switched systems, in Proc. Amer. Contr. Conf, vol 3, pp 1880–1885

Zhang W, Branicky MS (2001) Stability of networked control systems with time-varying transmission period. In: Allerton conference on communication, control, and computing, Monticello, IL

Zhang W, Branicky MS, Phillips SM (2001) Stability of networked control systems. IEEE Control Syst Mag 21(1):84–99

Networked Control Systems: Estimation and Control Over Lossy Networks

João P. Hespanha[1] and Alexandre R. Mesquita[2]
[1]Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA
[2]Department of Electronics Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

Abstract

This entry discusses optimal estimation and control for lossy networks. Conditions for stability are provided both for two-link and multiple-link networks. The online adaptation of network resources (controlled communication) is also considered.

Keywords

Automatic control; Communication networks; Controlled communication; Estimation; Networked control systems; Stability

Introduction

Network Control Systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators, and controllers

occurs through a shared band-limited digital communication network. In this entry, we consider the problem of estimation and control over such networks.

A significant difference between NCSs and standard digital control is the possibility that data may be lost while in transit through the network. Typically, *packet dropouts* result from transmission errors in physical network links (which is far more common in wireless than in wired networks) or from buffer overflows due to congestion. Long transmission delays sometimes result in packet reordering, which essentially amounts to a packet dropout if the receiver discards "outdated" arrivals. Reliable transmission protocols, such as TCP, guarantee the eventual delivery of packets. However, these protocols are not appropriate for NCSs since the retransmission of old data is generally not useful. Another important difference between NCSs and standard digital control systems is that, due to the nature of network traffic, delays in the control loop may be time varying and nondeterministic.

In this entry, we concentrate on the problem of control and estimation in the presence of packet losses, leaving other important features of NCSs (such as quantization and random delays) to be addressed in other entries of this encyclopedia. Consequently, we assume that the network can be viewed as a channel that can carry real numbers without distortion, but that some of the messages may be lost. This network model is appropriate when the number of bits in each data packet is sufficiently large so that quantization effects can be ignored, but packet dropouts cannot. For more general channel models, see, for example, Imer and Basar (2005).

This entry also does not address network transmission delays explicitly. In general, network delays have two components: one that is due to the time spent transmitting packets and another due to the time packets wait in buffers waiting to be transmitted. Delays due to packet transmission present little variation and may be modeled as constants. For control design purposes, these delays may be incorporated into the plant model. Delays due to buffering depend on the network

traffic and are typically random; they can be analyzed using the techniques developed in Antunes et al. (2012).

Notation and Basic Definitions. Throughout the entry, $\mathbb{R}$ stands for real numbers and $\mathbb{N}$ for nonnegative integers. For a given matrix $A \in \mathbb{R}^{n \times n}$ and vector $x \in \mathbb{R}^n$, $\|x\| := \sqrt{x'x}$ denotes the Euclidean norm of x, and $\lambda(A)$ the set of eigenvalues of A. Random variables are generally denoted in boldface. For a random variable $\mathbf{y}$, $\mathrm{E}[\mathbf{y}]$ stands for the expectation of $\mathbf{y}$.

Two-Link Networks

Here, we consider a control/estimation problem when all network effects can be modeled using two erasure channels: one from the sensor to the controller and the other from the controller to the actuator (see Fig. 1).

We restrict our attention to a linear time-invariant (LTI) plant with intermittent observation and control packets:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + \nu_k B \mathbf{u}_k + \mathbf{w}_k, \qquad (1a)$$

$$\mathbf{y}_k = \theta_k C \mathbf{x}_k + \mathbf{v}_k, \qquad (1b)$$

$\forall k \in \mathbb{N}, \mathbf{x}_k, \mathbf{w}_k \in \mathbb{R}^n, \mathbf{y}_k, \mathbf{v}_k \in \mathbb{R}^p$, where $(\mathbf{x}_0, \mathbf{w}_k, \mathbf{v}_k)$ are mutually independent, zero-mean Gaussian with covariance matrices (P_0, R_w, R_v), and $\theta_k, \nu_k \in \{0, 1\}$ are i.i.d. Bernoulli random variables with $\Pr\{\theta_k = 1\} = \bar{\theta}$ and $\Pr\{\nu_k = 1\} = \bar{\nu}$. The variable θ_k models the packet loss between sensor and controller, whereas ν_k models the packet loss between controller and actuator. When there is a packet drop from controller
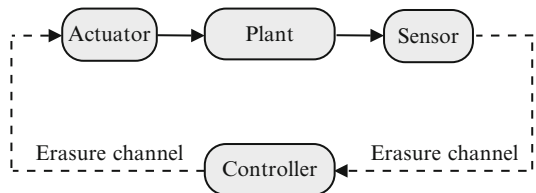
Networked Control Systems: Estimation and Control Over Lossy Networks, Fig. 1 Control system with two network links

to actuator, we set the actuator's output to zero. Different strategies, such as holding the control input, could still be modeled using (1) by augmenting of the state vector.

The information available to the controller up to time k is given by the information set:

$$\mathcal{I}_k = \{P_0\} \cup \{\mathbf{y}_\ell, \theta_\ell : \ell \leq k\} \cup \{\nu_\ell : \ell \leq k - 1\}.$$

Here, we make an important assumption that acknowledgment packets from the actuator are always received by the controller so that $\nu_\ell, \ell \leq k - 1$ is available at time k to the remote estimator.

Optimal Estimation with Remote Computation

The optimal mean-square estimate of $\mathbf{x}_k$, given the information known to the remote estimator at time k, is given by

$$\hat{\mathbf{x}}_{k|k} := \mathrm{E}[\mathbf{x}_k | \mathcal{I}_k].$$

This estimate can be computed recursively using the following time-varying Kalman filter (TVKF) (Sinopoli et al. 2004):

$$\hat{\mathbf{x}}_{0|-1} = 0, \tag{2a}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \theta_k F_k (\mathbf{y}_k - C\hat{\mathbf{x}}_{k|k-1}), \tag{2b}$$

$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k} + \nu_k B\mathbf{u}_k, \tag{2c}$$

with the gain matrix F_k calculated recursively as follows

$$F_k = P_k C'(CP_k C' + R_v)^{-1},$$

$$P_{k+1} = AP_k A' + R_w - \theta_k AF_k(CP_k C' + R_v) F_k' A'.$$

Each P_k corresponds to the estimation error covariance matrix

$$P_k = \mathrm{E}\left[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})'\right].$$

For this estimator, there exists a critical value θ_c for the dropout rate $\bar{\theta}$, above which the estimation error covariance becomes unbounded:

Theorem 1 (Sinopoli et al. 2004) *Assume that $\left(A, R_w^{1/2}\right)$ is controllable, (A, C) is observable, and A is unstable. Then there exists a critical value $\theta_c \in (0, 1]$ such that*

$$\mathrm{E}[P_k] \leq M, \forall k \in \mathbb{N} \Leftrightarrow \bar{\theta} \geq \theta_c$$

where M is a positive definite matrix that may depend on P_0. Furthermore, the critical value θ_c satisfies $\theta_{\min} \leq \theta_c \leq \theta_{\max}$, where the lower bound is given by

$$\theta_{\min} = 1 - \frac{1}{(\max\{|\lambda(A)|\})^2}, \tag{3}$$

and the upper bound is given by the solution to the following (quasi-convex) optimization problem:

$$\theta_{\max} = \min\{\theta \geq 0 : \Psi_\theta(Y, Z) > 0,$$
$$0 \leq Y \leq I \text{ for some } Y, Z\},$$

where

$$\Psi_\theta(Y, Z) =$$

$$\begin{bmatrix} Y & \sqrt{\theta}(YA + ZC) & \sqrt{1-\theta}YA \\ \sqrt{\theta}(A'Y + C'Z') & Y & 0 \\ \sqrt{1-\theta}A'Y & 0 & Y \end{bmatrix}.$$

Remark 1 In some special cases, the upper bound in (3) is tight in the sense that $\theta_c - \theta_{\min}$. The largest class of systems known for which this occurs is that of *nondegenerate systems* defined in Mo and Sinopoli (2012). Examples of systems in this class include (1) those for which the matrix C is invertible and (2) those with a detectable pair (A, C) and such that the matrix A is diagonalizable with unstable eigenvalues having distinct absolute values.

Optimal Control with Remote Computation

From a control perspective, one may also be interested in finding control sequences $\mathbf{u}^N = \{\mathbf{u}_1, \ldots, \mathbf{u}_{N-1}\}$, as functions of the information set $\mathcal{I}_N$, which minimize cost functions of the form

$$J = \lim_{N \to \infty} \frac{1}{N} \mathrm{E} \left[\sum_{k=0}^{N-1} (\mathbf{x}_k' W \mathbf{x}_k + \nu_k \mathbf{u}_k' U \mathbf{u}_k) | \mathcal{I}_k \right].$$

Theorem 2 (Schenato et al. 2007) *Assume that (A, B) and $(A, R_w^{1/2})$ are controllable, (A, C) and $(A, W^{1/2})$ are observable, and A is unstable. Then, finite control costs J are achievable if and only if $\bar{\theta} > \theta_c$ and $\bar{\nu} > \nu_c$, where the critical value ν_c is given by the (quasi-convex) optimization problem*

$$\nu_c = \min\{\nu \geq 0 : \Psi_\nu(Y, Z) > 0, \\ 0 \leq Y \leq I \text{ for some } Y, Z\},$$

where

$$\Psi_\nu(Y, Z) =$$
$$\begin{bmatrix} Y & Y & \sqrt{\nu} Z U^{1/2} & \sqrt{\nu}(YA' + ZB') & \sqrt{1-\nu}YA' \\ Y & W^{-1} & 0 & 0 & 0 \\ \sqrt{\nu}U^{1/2}Z' & 0 & I & 0 & 0 \\ \sqrt{\nu}(AY + BZ') & 0 & 0 & Y & 0 \\ \sqrt{1-\nu}AY & 0 & 0 & 0 & Y \end{bmatrix}.$$

Moreover, under the above conditions, the separation principle holds in the sense that the optimal control is given by

$$\mathbf{u}_k = -(B'SB + U)^{-1} B'SA\hat{\mathbf{x}}_{k|k},$$

where $\hat{\mathbf{x}}_{k|k}$ is an optimal state estimate given by (2) and the matrix S is the solution to the modified algebraic Riccati (MARE) equation

$$S = A'SA + W - \bar{\nu}A'SB(B'SB + U)^{-1}B'SA.$$

Solutions to the MARE may be obtained iteratively when $\bar{\nu} > \nu_c$.

Estimation with Local Computation

To reduce the gap between the bounds $\theta_{\min}$ and $\theta_{\max}$ on the critical value of the drop probability in Theorem 1 and to allow for larger probabilities of drop, one may choose to compute state estimates at the sensor and transmit those to the controller/actuator. This scheme is motivated by the growing number of *smart sensors* with embedded processing units that are capable of local computation. For the LTI plant

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k,$$
$$\mathbf{y}_k = C\mathbf{x}_k + \mathbf{v}_k,$$

the smart sensor can compute locally an optimal state estimate using a standard stationary Kalman filter and transmits this estimate to the controller. We model packet dropouts as before using the process θ_k and assume that the process θ_k is known to the smart sensor by means of an perfect acknowledgment mechanism. This allows the sensor to know $\mathbf{u}_k$ exactly and to use it in the Kalman filter.

Let $\tilde{\mathbf{x}}_{k|k} = \mathrm{E}[\mathbf{x}_k | \mathbf{y}_\ell, \theta_\ell, \ell \leq k]$ denote the local estimates transmitted by the sensor. Using the messages successfully received up to time k, the remote estimator computes the optimal estimate

$$\hat{\mathbf{x}}_{k|k-1} = \mathrm{E}[\mathbf{x}_k | \theta_\ell, \tilde{\mathbf{x}}_{\ell|\ell}, \ell \leq k-1].$$

recursively by

$$\hat{\mathbf{x}}_{0|-1} = 0,$$
$$\hat{\mathbf{x}}_{k|k} = (1 - \theta_k)\hat{\mathbf{x}}_{k|k-1} + \theta_k\tilde{\mathbf{x}}_{k|k}, \quad k \in \mathbb{N},$$
$$\hat{\mathbf{x}}_{k+1|k} = A\hat{\mathbf{x}}_{k|k} + B\mathbf{u}_k$$

N

Notice that now we are applying the (TVKF) to estimate $\tilde{\mathbf{x}}_k$, which is fully observable. Since $\theta_{\min}$ and $\theta_{\max}$ in Theorem 2 are equal for fully observable processes (Schenato et al. 2007), the local computation scheme grants a minimal critical value θ_c as stated in the theorem below.

Theorem 3 *Assume that $(A, R_w^{1/2})$ is controllable, (A, C) is* OBSERVABLE*, and A is unstable. Then the critical value θ_c is given by $\theta_{\min}$ in* (3)*, i.e.,*

$$\mathrm{E}[P_k] \leq M, \forall k \in \mathbb{N} \quad \Leftrightarrow \quad \bar{\theta} \geq \theta_{\min}$$

where M is a positive definite matrix that may depend on P_0.

Drops in the Acknowledgement Packets

When there are drops in the acknowledgment channel from the actuator to the controller, the controller does not always know ν_k, and therefore, it might not always have access to the control inputs that are actually applied to the plant. In this case, the posterior state probability becomes a Gaussian mixture distribution with infinitely many components, and the separation principle no longer holds (Schenato et al. 2007). This makes the estimation and control problems computationally more difficult, and, due to the smaller information set, some performance degradation in the control performance should be expected. For this reason, it is generally a good design choice to keep controller and actuator collocated when drops in the acknowledgment channels are significant.

Buffering

As an alternative to the approach described in section "Estimation with Local Computation" to use local computation at a smart sensor to allow for larger probabilities of drop, the designer may also consider the transmission of a sequence of previous measurements $\mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_{k-N}$ in each packet. This approach is motivated by the fact that often data packets can carry much more than one vector of measured outputs. When N is reasonably large, one should expect similar estimation/control performances as in

the approach described in section "Estimation with Local Computation", but with a reduced computational effort at the sensor.

Analogously, an improvement to zeroing or simply holding the control input in case of packet drops between controller and actuator is for the controller to transmit a control sequence $\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+N}$ that contains not only the control $\mathbf{u}_k$ to be used at the current time instant but also a few future controls $\mathbf{u}_{k+1}, \mathbf{u}_{k+2}, \ldots, \mathbf{u}_{k+N}$. In the case of packet drops between controller and actuator, the actuator can use previously received "future" control inputs in lieu of the one contained in the lost packet. The sequence of future control inputs may be obtained, e.g., by an optimal receding horizon control strategy (Gupta et al. 2006).

Estimation with Markovian Drops

When θ_k is a Markov process, we no longer have a separation principle, and the optimal controller may depend on the drops sequence. Yet, optimal state estimates are obtained using the same TVKF presented earlier. Below, we give conditions for the stability of the error covariance when drops are governed by the Gilbert-Elliot model: $\Pr\{\theta_{k+1} = j | \theta_k = i\} = p_{ij}, i, j \in \{0, 1\}$.

Theorem 4 (Mo and Sinopoli 2012) *Assume that $(A, R_w^{1/2})$ is controllable, A is unstable, and the system given by the pair (A, C) is nondegenerate as discussed in Remark 1. Moreover, suppose that the transition probabilities for the Gilbert-Elliot model satisfy $p_{01}; p_{10} > 0$. Then the expected error covariance $\mathrm{E}[P_k]$ is uniformly bounded if*

$$p_{01} > \theta_{\min}$$

and it is unbounded for some initial condition if $p_{01} < \theta_{\min}$.

Networks with Multiple Links

We now consider feedback loops that are closed over a network of communication links, each of which drops packets according to a Bernoulli

process. The sensor communicates with a controller across the network, and we assume that controller and actuator are collocated. The network may be represented by a graph $\mathcal{G}$ with nodes in the set $\mathcal{V}$ and edges in the set $\mathcal{E}$, where edges are drawn between two communicating nodes. We denote by p_{ij} the probability of a drop when node i transmits to node j. Drops are assumed to be independent across links and time.

To maximize robustness with respect to drops, sensors use a Kalman filter to compute an optimal estimate for the state of the process based on their measurements and transmit this estimate across the network. When the sensors do not have access to the process input, they can take advantage of the linearity of the Kalman filter: as the output of a Kalman filter is the sum of a term due to measurements with another term due to control inputs, sensors may compute only the contribution due to measurements and transmit it to the controller, which can subsequently add the contribution due to the control inputs. This guarantees that optimal state estimates can still be computed at the control node, even when the sensors do not know the control input (Gupta et al. 2009).

The communication in the network goes as follows. Sensors time stamp their estimates and broadcast them to all nodes in their communication ranges. After receiving information from their neighbors, nodes compare time stamps and keep only the most recent estimates. These estimates are broadcasted to all neighboring nodes. When the controller receives new information, the optimal Kalman estimate is reconstructed, taking into account the total transmission delay (learned from the packet time stamps), and a standard LQG control can be used (Gupta et al. 2009).

To determine whether or not this procedure results in a stable closed loop, one defines a *cut* $\mathcal{C} = (\mathcal{S}, \mathcal{T})$ to be a partition of the node set $\mathcal{V}$ such that the sensor node is in $\mathcal{S}$ and the controller node is in $\mathcal{T}$. The cut-set is then defined as the set of edges $(i, j) \in \mathcal{E}$ such that $i \in \mathcal{S}$ and $j \in \mathcal{T}$, i.e., the set of edges that connect the sets $\mathcal{S}$ and $\mathcal{T}$. The *max-cut probability* is then defined as

$$p_{\text{max-cut}} = \max_{\text{all cuts}(\mathcal{S},\mathcal{T})} \prod_{(i,j)\in\mathcal{S}\times\mathcal{T}} p_{ij}.$$

The above maximization can be rewritten as a minimization over the sums of $-\log p_{ij}$, which leads to a linear program known as the minimum cut problem in network optimization theory (Cook 1995).

Theorem 5 (Gupta et al. 2009) *Assume that* $R_w, R_v > 0$, *that* (A, B) *is stabilizable, that* (A, C) *is observable, and that A is unstable. Then the control and communication policy described above is optimal for quadratic costs, and the expected state covariance is bounded if and only if*

$$p_{\text{max-cut}} \cdot (\max\{|\lambda(A)|\})^2 < 1.$$

Estimation with Controlled Communication

To actively reduce network traffic and power consumption, sensor measurements may not be sent to the remote estimator at every time step. In addition, one may have the ability to somewhat control the probability of packet drops by varying the transmit power or by transmitting copies of the same message through multiple channel realizations. This is known as *controlled communication*, and it allows the designer to establish a trade-off between communication and estimation performance.

We consider the local estimation scenario described in section "Estimation with Local Computation" with the difference that the Bernoulli drops are now modulated as follows

$$\theta_k = \begin{cases} 1 & \text{with prob. } \Lambda_k \\ 0 & \text{with prob. } 1 - \Lambda_k \end{cases}$$

where the sensor is free to choose $\Lambda_k \in [0, p_{\text{max}}]$ as a function of the information available up to time k. With its choice, the sensor incurs on a communication cost $c(\Lambda_k)$ at time k, where $c(\cdot)$ is some increasing function that may represent, for example, the energy needed in order to transmit with a probability of drop equal to Λ_k. Note

that transmission scheduling, where Λ_k is either 0 or $p_{\max}$, is a special case of this framework.

In order to choose Λ_k, the sensor considers the estimation error $\tilde{\mathbf{e}}_k := \tilde{\mathbf{x}}_{k|k} - \hat{\mathbf{x}}_{k|k-1}$ between the local and the remote estimators. This error evolves according to

$$\tilde{\mathbf{e}}_{k+1} = \begin{cases} \mathbf{d}_k & \text{with prob. } \Lambda_k \\ A\tilde{\mathbf{e}}_k + \mathbf{d}_k & \text{with prob. } 1 - \Lambda_k \end{cases}$$

where $\mathbf{d}_k$ is the innovations process arising from the standard Kalman filter in the smart sensor.

Our objective is to find a "communication policy" that minimizes the long-term average cost

$$\tilde{J} := \lim_{K\to\infty} \frac{1}{K}\mathrm{E}\left[\sum_{k=0}^{K-1}\|\tilde{\mathbf{e}}_k\|^2 + \lambda c(\Lambda_k)\right],$$
$$\lambda > 0, \tag{4}$$

which penalizes a linear combination of the remote estimation error variance $\mathrm{E}\left[\|\tilde{\mathbf{e}}_k\|^2\right]$ and the average communication cost $\mathrm{E}[c(\Lambda_k)]$. In this context, a *communication policy* should be understood as a rule that selects Λ_k as a function of the information available to the sensor.

When

$$(1 - p_{\max})\max\{|\lambda(A)|\}^2 < 1,$$

there exists an optimal communication policy that chooses Λ_k as a function of $\tilde{e}_k$, which may be computed via dynamic programming and value iteration (Mesquita et al. 2012). While this procedure can be computationally difficult, it is often possible to obtain suboptimal but reasonable performance with rollout policies such as the following one:

$$\Lambda_k = \arg\min_{\Lambda \in [0, p_{\max}]}[(p_{\max} - \Lambda)\tilde{\mathbf{e}}_k' A' H A \tilde{\mathbf{e}}_k$$
$$+ \lambda c(\Lambda)] \tag{5}$$

where H is the positive semidefinite solution to the Lyapunov equation $(1 - p_{\max})A'HA - H = -I$ (Mesquita et al. 2012).

When computing $\tilde{\mathbf{e}}_k$ and Λ_k in (5) is computationally too costly for the sensor, one may prefer to make Λ_k a function of the number of consecutive dropped packets ℓ_k. In this case, minimizing $\tilde{J}$ in (4) is equivalent to minimizing the cost

$$\bar{J} := \lim_{K\to\infty} \frac{1}{K}\mathrm{E}\left[\sum_{k=0}^{K-1}\text{trace}\left(\Sigma_{\ell_k}\right) + \lambda c(\Lambda_k)\right],$$

where

$$\Sigma_\ell := \sum_{m=0}^{\ell} A'^m R_w A^m.$$

Since ℓ_k belongs to a countable set, one can very efficiently solve this optimization using dynamic programming (Mesquita et al. 2012).

Summary and Future Directions

Most positive results in the subject rely on the assumption of perfect acknowledgments and on actuators and controllers being collocated. Future research should address ways of circumventing these assumptions.

Cross-References

▶ Data Rate of Nonlinear Control Systems and Feedback Entropy
▶ Information and Communication Complexity of Networked Control Systems
▶ Networked Control Systems: Architecture and Stability Issues
▶ Networked Systems
▶ Quantized Control and Data Rate Constraints

Bibliography

Antunes D, Hespanha JP, Silvestre C (2012) Volterra integral approach to impulsive renewal systems: application to networked control. IEEE Trans Autom Control 57:607–619

Cook WJ (1995) Combinatorial optimization: papers from the DIMACS special year, vol 20. American Mathematical Society, Providence

Gupta V, Sinopoli B, Adlakha S, Goldsmith A, Murray R (2006) Receding horizon networked control. In: Proceedings of the allerton conference on communication, control, and computing, Monticello

Gupta V, Dana AF, Hespanha JP, Murray RM, Hassibi B (2009) Data transmission over networks for estimation and control. IEEE Trans Autom Control 54(8):1807–1819

Imer OC, Basar T (2005) Optimal estimation with limited measurements. In: Proceedings of the 44th IEEE conference on decision and control, 2005 and 2005 European control conference (CDC-ECC'05), Seville

Mesquita AR, Hespanha JP, Nair GN (2012) Redundant data transmission in control/estimation over lossy networks. Automatica 48(8):1612–1620

Mo Y, Sinopoli B (2012) Kalman filtering with intermittent observations: tail distribution and critical value. IEEE Trans Autom Control 57(3):677–689

Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry SS (2007) Foundations of control and estimation over lossy networks. Proc IEEE 95(1):163–187

Sinopoli B, Schenato L, Franceschetti M, Poolla K, Jordan MI, Sastry SS (2004) Kalman filtering with intermittent observations. IEEE Trans Autom Control 49(9):1453–1464

Networked Systems

Jorge Cortés

Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA, USA

Abstract

This entry provides a brief overview on networked systems from a systems and control perspective. We pay special attention to the nature of the interactions among agents; the critical role played by information sharing, dissemination, and aggregation; and the distributed control paradigm to engineer the behavior of networked systems.

Keywords

Autonomous networks; Cooperative control; Multi-agent systems; Swarms

Introduction

Networked systems appear in numerous scientific and engineering domains, including communication networks (Toh 2001), multi-robot networks (Arkin 1998; Balch and Parker 2002), sensor networks (Santi 2005; Schenato et al. 2007), water irrigation networks (Cantoni et al. 2007), power and electrical networks (Chow 1982; Chiang et al. 1995; Dörfler et al. 2013), camera networks (Song et al. 2011), transportation networks (Ahuja et al. 1993), social networks (Jackson 2010), and chemical and biological networks (Kuramoto 1984; Strogatz 2003). Their applications are pervasive, ranging from environmental monitoring, ocean sampling, and marine energy systems, through search and rescue missions, high-stress deployment in disaster recovery, health monitoring of critical infrastructure to science imaging, the smart grid, and cybersecurity.

The rich nature of networked systems makes it difficult to provide a definition that, at the same time, is comprehensive enough to capture their variety and simple enough to be expressive of their main features. With this in mind, we loosely define a networked system as a "system of systems," i.e., a collection of agents that interact with each other. These groups might be heterogeneous, composed by human, biological, or engineered agents possessing different capabilities regarding mobility, sensing, actuation, communication, and computation. Individuals may have objectives of their own or may share a common objective with others – which in turn might be adversarial with respect to another subset of agents.

In a networked system, the evolutions of the states of individual agents are coupled. Coupling might be the result of the physical interconnection among the agents, the consequence of the implementation of coordination algorithms where agents use information about each other, or a combination of both. There is diversity too in the nature of agents themselves and the interactions among them, which might be cooperative, adversarial, or belong to the rich range between the two. Due to changes in the state of the agents, the network, or the environment, interactions among

agents may be changing and dynamic. Such interactions may be structured across different layers, which themselves might be organized in a hierarchical fashion. Networked systems may also interact with external entities that specify high-level commands that trickle down through the system all the way to the agent level.

A defining characteristic of a networked system is the fact that information, understood in a broad sense, is sparse and distributed across the agents. As such, different individuals have access to information of varying degrees of quality. As part of the operation of the networked system, mechanisms are in place to share, transmit, and/or aggregate this information. Some information may be disseminated throughout the whole network or, in some cases, all information can be made centrally available at a reasonable cost. In other scenarios, however, the latter might turn out to be too costly, unfeasible, or undesirable because of privacy and security considerations. Individual agents are the basic unit for decision making, but decisions might be made from intermediate levels of the networked system all the way to a central planner. The combination of information availability and decision-making capabilities gives rise to an ample spectrum of possibilities between the centralized control paradigm, where all information is available at a central planner who makes the decisions, and the fully distributed control paradigm, where individual agents only have access to the information shared by their neighbors in addition to their own.

Perspective from Systems and Control

There are many aspects that come into play when dealing with networked systems regarding computation, processing, sensing, communication, planning, motion control, and decision making. This complexity makes their study challenging and fascinating and explains the interest that, with different emphases, they generate in a large number of disciplines. In biology, scientists analyze synchronization phenomena and self-organized swarming behavior in groups with

distributed agent-to-agent interactions (Okubo 1986; Parrish et al. 2002; Conradt and Roper 2003; Couzin et al. 2005). In robotics, engineers design algorithmic solutions to help multivehicle networks and embedded systems coordinate their actions and perform challenging spatially distributed tasks (Arkin 1998; Committee on Networked Systems of Embedded Computers 2001; Balch and Parker 2002; Howard et al. 2006; Kumar et al. 2008). Graph theorists and applied mathematicians study the role played by the interconnection among agents in the emergence of phase transition phenomena (Bollobás 2001; Meester and Roy 2008; Chung 2010). This interest is also shared in communication and information theory, where researchers strive to design efficient communication protocols and examine the effect of topology control on group connectivity and information dissemination (Zhao and Guibas 2004; Giridhar and Kumar 2005; Lloyd et al. 2005; Santi 2005; Franceschetti and Meester 2007). Game theorists study the gap between the performance achieved by global, network-wide optimizers and the configurations that result from selfish agents interacting locally in social and economic systems (Roughgarden 2005; Nisan et al. 2007; Easley and Kleinberg 2010; Marden and Shamma 2013). In mechanism design, researchers seek to align the objectives of individual self-interested agents with the overall goal of the network. Static and mobile networked systems and their applications to the study of natural phenomena in oceans (Paley et al. 2008; Graham and Cortés 2012; Zhang and Leonard 2010; Das et al. 2012; Ouimet and Cortés 2013), rivers (Ru and Martínez 2013; Tinka et al. 2013), and the environment (DeVries and Paley 2012) also raise exciting challenges in estimation theory, computational geometry, and spatial statistics.

The field of systems and control brings a comprehensive approach to the modeling, analysis, and design of networked systems. Emphasis is put on the understanding of the general principles that explain how specific collective behaviors emerge from basic interactions; the establishment of models, abstractions, and tools that allow us to reason rigorously about complex

interconnected systems; and the development of systematic methodologies that help engineer their behavior. The ultimate goal is to establish a science for integrating individual components into complex, self-organizing networks with predictable behavior. To realize the "power of many" and expand the realm of what is possible to achieve beyond the individual agent capabilities, special care is taken to obtain precise guarantees on the stability properties of coordination algorithms, understand the conditions and constraints under which they work, and characterize their performance and robustness against a variety of disturbances and disruptions.

Research Issues – and How the Entries in the Encyclopedia Address Them

Given the key role played by agent-to-agent interactions in networked systems, the Encyclopedia entries ▶ Graphs for Modeling Networked Interactions and ▶ Dynamic Graphs, Connectivity of deal with how their nature and effect can be modeled through graphs. This includes diverse aspects such as deterministic and stochastic interactions, static and dynamic graphs, state-dependent and time-dependent neighboring relationships, and connectivity. The importance of maintaining a certain level of coordination and consistency across the networked system is manifested in the various entries that deal with coordination tasks that are, in some way or another, related to some form of agreement. These include consensus (▶ Averaging Algorithms and Consensus), formation control (▶ Vehicular Chains), cohesiveness, flocking (▶ Flocking in Networked Systems), synchronization (▶ Oscillator Synchronization), and distributed optimization (▶ Distributed Optimization). A great deal of work (e.g., see ▶ Optimal Deployment and Spatial Coverage and ▶ Multi-vehicle Routing), is also devoted to the design of cooperative strategies that achieve spatially distributed tasks such as optimal coverage, space partitioning, vehicle routing, and servicing. These entries explore the optimal placement of agents,

the optimal tuning of sensors, and the distributed optimization of network resources. The entry ▶ Estimation and Control over Networks explores the impact that communication channels may have on the execution of estimation and control tasks over networks of sensors and actuators. A strong point of commonality among the contributions is the precise characterization of the scalability of coordination algorithms, together with the rigorous analysis of their correctness and stability properties. Another focal point is the analysis of the performance gap between centralized and distributed approaches in regard to the ultimate network objective.

Further information about other relevant aspects of networked systems can be found throughout this Encyclopedia. Among these, we highlight the synthesis of cooperative strategies for data fusion, distributed estimation, and adaptive sampling, the analysis of the network operation under communication constraints (e.g., limited bandwidth, message drops, delays, and quantization), the treatment of game-theoretic scenarios that involve interactions among multiple players and where security concerns might be involved, distributed model predictive control, and the handling of uncertainty, imprecise information, and events via discrete-event systems and triggered control.

Summary and Future Directions

In conclusion, this entry has illustrated ways in which systems and control can help us design and analyze networked systems. We have focused on the role that information and agent interconnection play in shaping their behavior. We have also made emphasis on the increasingly rich set of methods and techniques that allow to provide correctness and performance guarantees. The field of networked systems is vast and the amount of work impossible to survey in this brief entry. The reader is invited to further explore additional topics beyond the ones mentioned here. The monographs (Ren and Beard 2008; Bullo et al. 2009; Mesbahi and Egerstedt 2010; Alpcan and Başar 2010) and edited

volumes (Kumar et al. 2004; Shamma 2008; Saligrama 2008), and manuscripts (Olfati-Saber et al. 2007; Baillieul and Antsaklis 2007; Leonard et al. 2007; Kim and Kumar 2012), together with the references provided in the Encyclopedia entries mentioned above, are a good starting point to undertake this enjoyable effort. Given the big impact that networked systems have, and will continue to have, in our society, from energy and transportation, through human interaction and healthcare, to biology and the environment, there is no doubt that the coming years will witness the development of more tools, abstractions, and models that allow to reason rigorously about intelligent networks and for techniques that help design truly autonomous and adaptive networks.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Dynamic Graphs, Connectivity of
▶ Distributed Optimization
▶ Estimation and Control over Networks
▶ Flocking in Networked Systems
▶ Graphs for Modeling Networked Interactions
▶ Multi-vehicle Routing
▶ Optimal Deployment and Spatial Coverage
▶ Oscillator Synchronization
▶ Vehicular Chains

Bibliography

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms, and applications. Prentice Hall, Englewood Cliffs

Alpcan T, Başar T (2010) Network security: a decision and game-theoretic approach. Cambridge University Press, Cambridge, UK

Arkin RC (1998) Behavior-based robotics. MIT, Cambridge, MA

Baillieul J, Antsaklis PJ (2007) Control and communication challenges in networked real-time systems. Proc IEEE 95(1):9–28

Balch T, Parker LE (eds) (2002) Robot teams: from diversity to polymorphism. A. K. Peters, Wellesley, MA

Bollobás B (2001) Random graphs, 2nd edn. Cambridge University Press, Cambridge, UK

Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Applied mathematics series. Princeton University Press, Princeton, NJ. Electronically available at http://coordinationbook.info

Cantoni M, Weyer E, Li Y, Ooi SK, Mareels I, Ryan M (2007) Control of large-scale irrigation networks. Proc IEEE 95(1):75–91

Chiang HD, Chu CC, Cauley G (1995) Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective. Proc IEEE 83(11):1497–1529

Chow JH (1982) Time-scale modeling of dynamic networks with applications to power systems. Springer, New York, NY

Chung FRK (2010) Graph theory in the information age. Not AMS 57(6):726–732

Committee on Networked Systems of Embedded Computers (2001) Embedded, everywhere: a research agenda for networked systems of embedded computers. National Academy Press, Washington, DC

Conradt L, Roper TJ (2003) Group decision-making in animals. Nature 421(6919):155–158

Couzin ID, Krause J, Franks NR, Levin SA (2005) Effective leadership and decision-making in animal groups on the move. Nature 433(7025):513–516

Das J, Py F, Maughan T, O'Reilly T, Messié M, Ryan J, Sukhatme GS, Rajan K (2012) Coordinated sampling of dynamic oceanographic features with AUVs and drifters. Int J Robot Res 31(5):626–646

DeVries L, Paley D (2012) Multi-vehicle control in a strong flowfield with application to hurricane sampling. AIAA J Guid Control Dyn 35(3): 794–806

Dörfler F, Chertkov M, Bullo F (2013) Synchronization in complex oscillator networks and smart grids. Proc Natl Acad Sci 110(6):2005–2010

Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, Cambridge, UK

Franceschetti M, Meester R (2007) Random networks for communication. Cambridge University Press, Cambridge, UK

Giridhar A, Kumar PR (2005) Computing and communicating functions over sensor networks. IEEE J Sel Areas Commun 23(4):755–764

Graham R, Cortés J (2012) Adaptive information collection by robotic sensor networks for spatial estimation. IEEE Trans Autom Control 57(6):1404–1419

Howard A, Parker LE, Sukhatme GS (2006) Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment, and detection. Int J Robot Res 25(5–6):431–447

Jackson MO (2010) Social and economic networks. Princeton University Press, Princeton, NJ

Kim KD, Kumar PR (2012) Cyberphysical systems: a perspective at the centennial. Proc IEEE 100(Special Centennial Issue):1287–1308

Kumar V, Leonard NE, Morse AS (eds) (2004) Cooperative control. Lecture notes in control and information sciences, vol 309. Springer, New York, NY

Kumar V, Rus D, Sukhatme GS (2008) Networked robots. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, New York, NY, pp 943–958

Kuramoto Y (1984) Chemical oscillations, waves, and turbulence. Springer, New York, NY

Leonard NE, Paley D, Lekien F, Sepulchre R, Fratantoni DM, Davis R (2007) Collective motion, sensor networks and ocean sampling. Proc IEEE 95(1): 48–74

Lloyd EL, Liu R, Marathe MV, Ramanathan R, Ravi SS (2005) Algorithmic aspects of topology control problems for ad hoc networks. Mobile Netw Appl 10(1–2):19–34

Marden JR, Shamma JS (2013) Game theory and distributed control. In: Young P, Zamir S (eds) Handbook of game theory, vol 4. Elsevier, Oxford, UK

Meester R, Roy R (2008) Continuum percolation. Cambridge University Press, Cambridge, UK

Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Applied mathematics series. Princeton University Press, Princeton, NJ

Nisan N, Roughgarden T, Tardos E, Vazirani VV (2007) Algorithmic game theory. Cambridge University Press, Cambridge, UK

Okubo A (1986) Dynamical aspects of animal grouping: swarms, schools, flocks and herds. Adv Biophys 22:1–94

Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1):215–233

Ouimet M, Cortés J (2013) Collective estimation of ocean nonlinear internal waves using robotic underwater drifters. IEEE Access 1:418–427

Paley D, Zhang F, Leonard N (2008) Cooperative control for ocean sampling: the glider coordinated control system. IEEE Trans Control Syst Technol 16(4):735–744

Parrish JK, Viscido SV, Grunbaum D (2002) Self-organized fish schools: an examination of emergent properties. Biol Bull 202:296–305

Ren W, Beard RW (2008) Distributed consensus in multi-vehicle cooperative control. Communications and control engineering. Springer, New York, NY

Roughgarden T (2005) Selfish routing and the price of anarchy. MIT, Cambridge, MA

Ru Y, Martínez S (2013) Coverage control in constant flow environments based on a mixed energy-time metric. Automatica 49(9):2632–2640

Saligrama V (ed) (2008) Networked sensing information and control. Springer, New York, NY

Santi P (2005) Topology control in wireless ad hoc and sensor networks. Wiley, New York, NY

Schenato L, Sinopoli B, Franceschetti M, Poolla K, Sastry SS (2007) Foundations of control and estimation over lossy networks. Proc IEEE 95(1):163–187

Shamma JS (ed) (2008) Cooperative control of distributed multi-agent systems. Wiley, New York, NY

Song B, Ding C, Kamal AT, Farrel JA, Roy-Chowdhury AK (2011) Distributed camera networks. IEEE Signal Process Mag 28(3):20–31

Strogatz SH (2003) SYNC: the emerging science of spontaneous order. Hyperion, New York, NY

Tinka A, Rafiee M, Bayen A (2013) Floating sensor networks for river studies. IEEE Syst J 7(1):36–49

Toh CK (2001) Ad hoc mobile wireless networks: protocols and systems. Prentice Hall, Englewood Cliffs, NJ

Zhang F, Leonard NE (2010) Cooperative filters and control for cooperative exploration. IEEE Trans Autom Control 55(3):650–663

Zhao F, Guibas L (2004) Wireless sensor networks: an information processing approach. Morgan-Kaufmann, San Francisco, CA

Neural Control and Approximate Dynamic Programming

Frank L. Lewis[1] and Kyriakos G. Vamvoudakis[2]
[1]Arlington Research Institute, University of Texas, Fort Worth, TX, USA
[2]Center for Control, Dynamical Systems and Computation (CCDC), University of California, Santa Barbara, CA, USA

Abstract

There has been great interest recently in "universal model-free controllers" that do not need a mathematical model of the controlled plant, but mimic the functions of biological processes to learn about the systems they are controlling online, so that performance improves automatically. Neural network (NN) control has had two major thrusts: approximate dynamic programming, which uses NN to approximately solve the optimal control problem, and NN in closed-loop feedback control.

Keywords

Adaptive control; Learning systems; Neural networks; Optimal control; Reinforcement learning

Neural Feedback Control

The objective is to design NN feedback controllers that cause a system to follow, or track,

a prescribed trajectory or path. Consider the dynamics of an n-link robot manipulator

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \tag{1}$$

with $q(t) \in \mathbb{R}^n$ the joint variable vector, $M(q)$ an inertia matrix, V_m a centripetal/coriolis matrix, $G(q)$ a gravity vector, and $F(\cdot)$ representing friction terms. Bounded unknown disturbances and modeling errors are denoted by τ_d and the control input torque is $\tau(t)$. The sliding mode control approach (Slotine and Li 1987) can be generalized to NN control systems. Given a desired trajectory, $q_d \in \mathbb{R}^n$ define the tracking error $e(t) = q_d(t) - q(t)$ and the sliding variable error $r = \dot{e} + \lambda e$ with $\lambda = \lambda^T > 0$. Define the nonlinear robot function,

$$f(x) = M(q)(\ddot{q}_d + \lambda \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \lambda e) \\ + G(q) + F(\dot{q})$$

with the known vector $x(t)$ of measured signals is selected as, $x = \begin{bmatrix} e^T & \dot{e}^T & q_d^T & \dot{q}_d^T & \ddot{q}_d^T \end{bmatrix}$.

NN Controller for Continuous-Time Systems

The NN controller is designed based on *functional approximation properties* of NN as shown in Lewis et al. (1999). Thus, assume that $f(x)$ can be approximated by $\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x)$ with $\hat{V}, \hat{W}$ the estimated NN weights. Select the control input, $\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v$ with K_v a symmetric positive definite gain and $v(t)$ a robustifying function. This NN control structure is shown in Fig. 1. The outer proportional-derivative (PD) tracking loop guarantees robust behavior. The inner loop containing the NN is known as a feedback linearization loop, and the NN effectively learns the unknown dynamics online to cancel the nonlinearities of the system. Let the estimated sigmoid Jacobian be $\hat{\sigma}' \equiv \frac{d\sigma(z)}{dz}\big|_{z=\hat{V}^T x}$. Then, the NN weight tuning laws are provided by

$$\dot{\hat{W}} = F\hat{\sigma}r^T - F\hat{\sigma}'\hat{V}^T x r^T - kF \|r\| \hat{W},$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'\hat{W}r)^T - kG \|r\| \hat{V},$$

with any constant symmetric matrices $F, G > 0$, and scalar tuning parameter $k > 0$.

NN Controller for Discrete-Time Systems

Most feedback controllers today are implemented on digital computers. This requires the specification of control algorithms in discrete time or digital form (Lewis et al. 1999). To design such controllers, one may consider the discrete-time dynamics $x_{k+1} = f(x_k) + g(x_k)u_k$ with unknown functions $f(\cdot), g(\cdot)$. The digital NN controller derived in this situation has the form of a feedback linearization controller shown in Fig. 1. One can derive tuning algorithms, for a discrete-time neural network controller with L layers, that guarantee system stability and robustness (Lewis et al. 1999). For the i-th layer, the weight updates are of the form

$$\hat{W}_i(k+1) = \hat{W}_i(k) - \alpha_i \hat{\phi}_i(k) \hat{y}_i^T(k) \\ - \Gamma \left\| I - \alpha_i \hat{\phi}_i(k) \hat{\phi}_i(k)^T \right\| \hat{W}_i(k)$$

where $\hat{\phi}_i(k)$ are the output functions of layer i, $0 < \Gamma < 1$ is a design parameter, and

$$\hat{y}_i(k) = \begin{cases} \hat{W}_i^T \hat{\phi}_i(k) + K_v r(k) & \text{for } i = 1, \dots, L-1, \\ r(k+1) & \text{for } i = L \end{cases}$$

with $r(k)$ a filtered error.

Feedforward Neurocontroller

Industrial, aerospace, DoD, and MEMS assembly systems have actuators that generally contain deadzone, backlash, and hysteresis. Since these actuator nonlinearities appear in the feedforward loop, the NN compensator must also appear in the feedforward loop. This design is significantly more complex than for feedback NN controllers. Details are given in Lewis et al. (2002). Feedforward controllers can offset the effects of deadzone if properly designed. It can be shown that a NN deadzone compensator has the structure shown in Fig. 2.
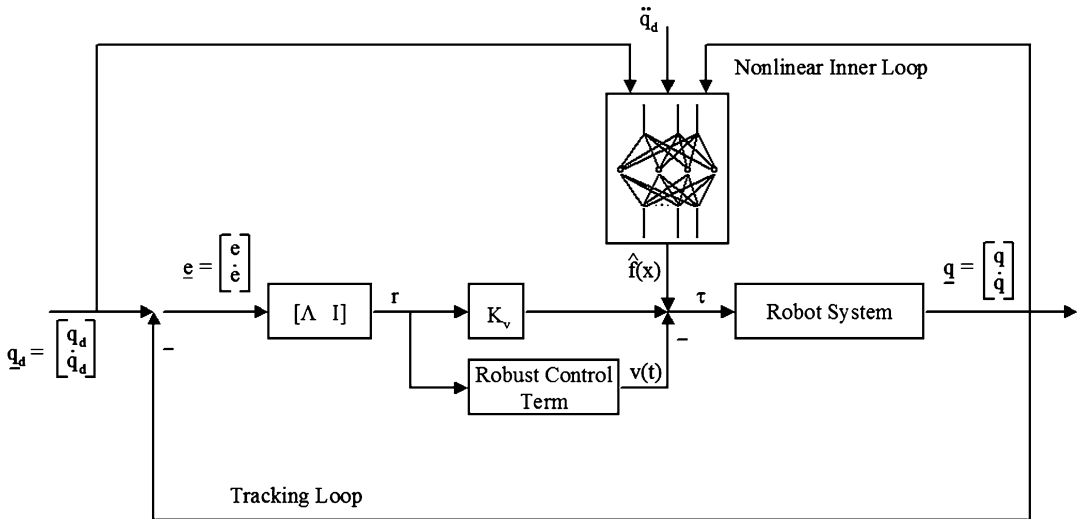
Neural Control and Approximate Dynamic Programming, Fig. 1 Neural network robot controller
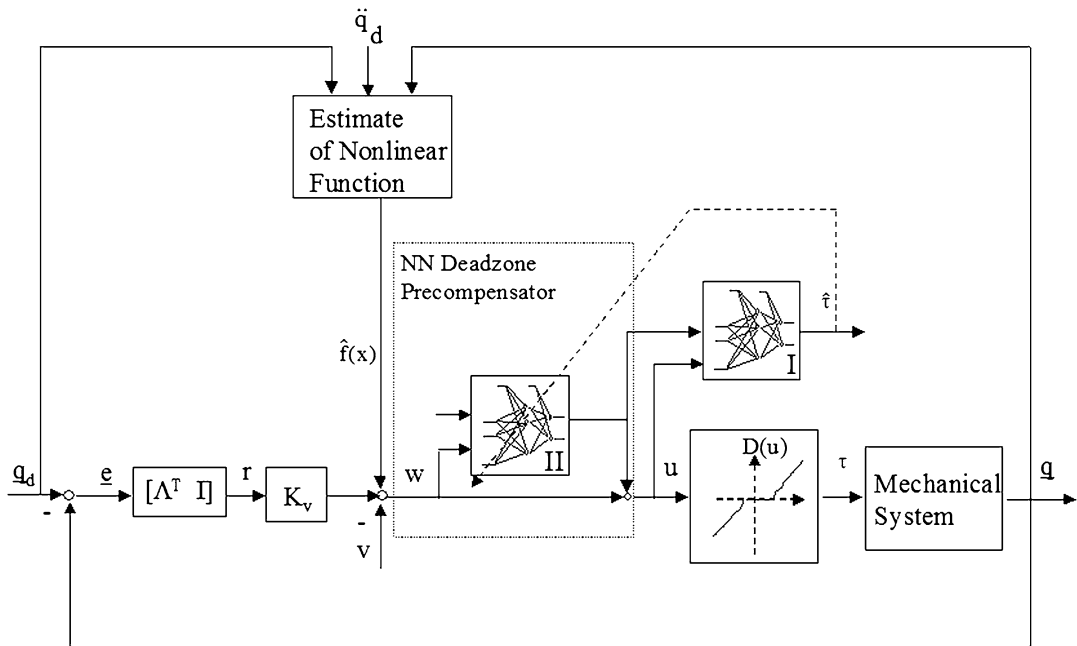
Neural Control and Approximate Dynamic Programming, Fig. 2 Feedforward NN for deadzone compensation

The NN compensator consists of two NNs. NN II is in the direct feedforward control loop, and NN I is not directly in the control loop but serves as an observer to estimate the (unmeasured) applied torque $\tau(t)$. The feedback stability and performance of the NN deadzone compensator have been rigorously proven using nonlinear stability proof techniques. The two NN were each selected as having one tunable layer, namely, the output weights. The activation functions were set as a basis by selecting fixed random values for the first-layer weights. To guarantee stability, the output weights of the inversion NN II (subscript i denotes weights and sigmoids of the inversion)

and the estimator NN I should be tuned respectively as

$$\dot{\hat{W}}_i = T\sigma_i(V_i w)r^T \hat{W}^T \sigma'(V^T u)V^T$$

$$- k_1 T \|r\| \hat{W}_i - k_2 T \|r\| \left\| \hat{W}_i \right\| \hat{W}_i,$$

$$\dot{\hat{W}} = -S\sigma'(V^T u)V^T \hat{W}_i \sigma_i(V_i^T w)r^T$$

$$- k_1 S \|r\| \hat{W},$$

with design matrices $T, S > 0$ and tuning gains k_1, k_2.

Approximate Dynamic Programming for Feedback Control

The current status of work in approximate dynamic programming (ADP) for feedback control is given in Lewis and Liu (2012). ADP is a form of reinforcement learning based on an actor/critic structure. Reinforcement learning (RL) is a class of methods used in machine learning to methodically modify the actions of an agent based on observed responses from its environment (Sutton and Barto 1998). The actor/critic structures are RL systems that have two learning structures: A critic network evaluates the performance of a current action policy, and based on that evaluation, an actor structure updates the action policy as shown in Fig. 3. Adaptive optimal controllers (Lewis et al. 2012b) have been proposed by adding optimality criteria to an adaptive controller or adding adaptive characteristics to an optimal controller.

Optimal Adaptive Control of Discrete-Time Nonlinear Systems

Consider a class of discrete-time systems described by the deterministic nonlinear dynamics in the affine state space difference equation form

$$x_{k+1} = f(x_k) + g(x_k)u_k, \qquad (2)$$

with state $x_k \in \mathbb{R}^n$ and control input $u_k \in \mathbb{R}^m$. A deterministic control policy is defined as a function from state space to control space $\mathbb{R}^n \to \mathbb{R}^m$. That is, for every state x_k, the policy defines a control action $u_k = h(x_k)$ as a feedback controller. Define a deterministic cost function that yields the value function:

$$V(x_k) = \sum_{i=k}^{\infty} \gamma^{i-k} r(x_i, u_i),$$

with $0 < \gamma \leq 1$ a discount factor, $Q(x_k)$, $R > 0$, and $u_k = h(x_k)$ a prescribed feedback control policy. The optimal value is given by Bellman's optimality equation:

$$V^*(x_k) = \min_{h(.)} \left(r(x_k, h(x_k)) + \gamma V^*(x_{k+1}) \right),$$

which is the discrete-time Hamilton-Jacobi-Bellman (HJB) equation. Two forms of RL can be based on policy iteration (PI) and value iteration (VI). For temporal difference learning, PI is

Neural Control and Approximate Dynamic Programming, Fig. 3 RL with an actor/critic structure
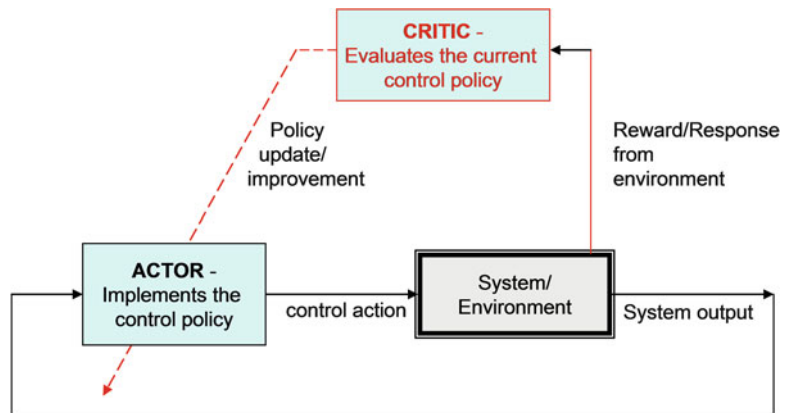
written as follows in terms of the deterministic Bellman equation.

Algorithm 1 PI for discrete-time systems
___
1: **procedure**
2:     Given admissible policies $h_0(x_k)$
3:     **while** $\left\| V^{h_i} - V^{h_i} \right\| \geq \epsilon_{ac}$ **do**
4:         Solve for the value $V_{(i)}(x)$ using

$$V_{i+1}(x_k) = r(x_k, h_i(x_k)) + \gamma V_{i+1}(x_{k+1})$$

5:         Update the control policy $h_{(i+1)}(x_k)$ using

$$h_{i+1}(x_k) = \arg\min_{h(\cdot)} \left(r(x_k, h(x_k)) + \gamma V_{i+1}(x_{k+1}) \right)$$

6:         $i := i + 1$
7:     **end while**
8: **end procedure**

---

where ϵ_{ac} is a small number that checks the algorithm convergence. Value iteration is similar, but the policy evaluation procedure is performed as $V_{i+1}(x_k) = r(x_k, h_i(x_k)) + \gamma V_i(x_{k+1})$. In value iteration, we can select any initial control policy, not necessarily admissible or stabilizing. In the control system shown in Fig. 3, the critic and the actor NNs are tuned online using the observed data $\left(x_k, x_{k+1}, r(x_k, h_i(x_k)) \right)$ along the system trajectory. The critic and actor are tuned sequentially in both the PI and the VI. That is, the weights of one neural network are held constant, while the weights of the other are tuned until convergence. This procedure is repeated until both neural networks have converged. Thus, the controller learns the optimal controller online. The convergence of value iteration using two neural networks for the discrete-time nonlinear system (2) is proven in Al-Tamimi et al. (2008). Design of an ADP controller that uses only output feedback is given in Lewis and Vamvoudakis (2011).

Optimal Adaptive Control of Continuous-Time Nonlinear Systems

RL is considerably more difficult for continuous-time systems than for discrete-time systems, and fewer results are available. This subsection will provide the formulation of optimal control problem followed by an offline PI algorithm provided

in Abu-Khalaf and Lewis (2005) that will give us the structure for the proposed online algorithms that follow. Consider the following nonlinear time-invariant affine in the input dynamical system given by

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t); x(0) = x_0 \quad (3)$$

with $x(t) \in \mathbb{R}^n$, $f(x(t)) \in \mathbb{R}^n$, $g(x(t)) \in \mathbb{R}^{n \times m}$ and control input $u(t) \in \mathbb{R}^m$. We assume that $f(0) = 0$, $f(x) + g(x)u$ is Lipschitz continuous on a set $\Omega \subset \mathbb{R}^n$ that contains the origin and that the system is stabilizable on Ω, that is, there exists a continuous control function $u(t) \in U$ such that the system is asymptotically stable on Ω. Define the infinite horizon integral cost $\forall t \geq 0$

$$V(x_t) = \int_t^\infty r(x(\tau), u(\tau)) d\tau, \quad (4)$$

with $Q(x)$ positive definite and $R \in \mathbb{R}^{m \times m}$ a symmetric positive definite matrix. For any admissible control policy if the associated cost (4) is C^1, then an infinitesimal version is the Bellman equation, and the optimal cost function $V^*(x)$ is defined by

$$V^*(x_0) = \min_u \left(\int_0^\infty r(x, u) d\tau \right)$$

which satisfies the HJB equation. By employing the stationarity condition, the optimal control function for the given problem is

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^*(x)}{\partial x}. \quad (5)$$

Inserting the optimal control (5) into the Bellman equation, one obtains the formulation for HJB equation in terms of $\frac{\partial V^*(x)}{\partial x}$ and with boundary condition $V^*(0) = 0$

$$0 = r(x, u^*) + \frac{\partial V^*(x)}{\partial x}^T (f(x) + g(x)u^*), \quad (6)$$

which for the linear case becomes the well-known Riccati equation. In order to find the

optimal control solution for the problem, one needs to solve the HJB equations (6) for the value function and then substitute in (5) to obtain the optimal control. However, due to the nonlinear nature of the HJB equation, finding its solution is generally difficult or impossible. The following PI algorithm is an iterative algorithm for solving optimal control problems and will give us the structure for the online learning algorithm.

Algorithm 2 PI for continuous-time systems

1: **procedure**
2:     Given admissible policies $u^{(0)}$
3:     **while** $\left\| V^{u^{(i)}} - V^{u^{(i)}} \right\| \geq \epsilon_{ac}$ **do**
4:         Solve for the value $V^{(i)}(x)$ using Bellman's equation

$$Q(x) + \frac{\partial V^{u^{(i)}}}{\partial x}^T (f(x) + g(x)u^{(i)}) + u^{(i)^T} Ru^{(i)} = 0,$$

$$V^{u^{(i)}}(0) = 0$$

5:         Update the control policy $u^{(i+1)}$ using

$$u^{(i+1)} = -\left(\frac{1}{2} R^{-1} g^T(x) \frac{\partial V^{u^{(i)}}}{\partial x}^T \right)$$

6:         $i := i + 1$
7:     **end while**
8: **end procedure**

A PI algorithm that solves online the HJB equation without full information of the plant dynamics is proposed in Vrabie et al. (2009) where the Bellman equation is proved to be equivalent to the *integral reinforcement learning form* with an optimal value given for some $T > 0$ as

$$V^*(x(t)) = \arg \min_u \int_t^{t+T} r(x(\tau), u(\tau)) d\tau + V^*(x(t+T)).$$

Therefore, the temporal difference error for continuous-time systems can be defined as

$$e(t : t + T) = \rho(t : t + T) + V(x(t+T)) - V(x(t)),$$

with $\rho(t : t + T) \equiv \int_t^{t+T} r(x(\tau), u(\tau)) d\tau$ without any information of the plant dynamics. The IRL controller just given tunes the critic neural network to determine the value while holding the control policy fixed. The IRL algorithm can be implemented online by RL techniques using value function approximation $\hat{V}(x) = \hat{W}_1^T \phi(x)$ in a critic approximator network. Using that approximation in the PI algorithm, one can use batch least squares or recursive least squares to update the value function, and then on convergence of the value parameters, the action is updated. The implementation of the IRL optimal adaptive control algorithm is shown in Fig. 4.
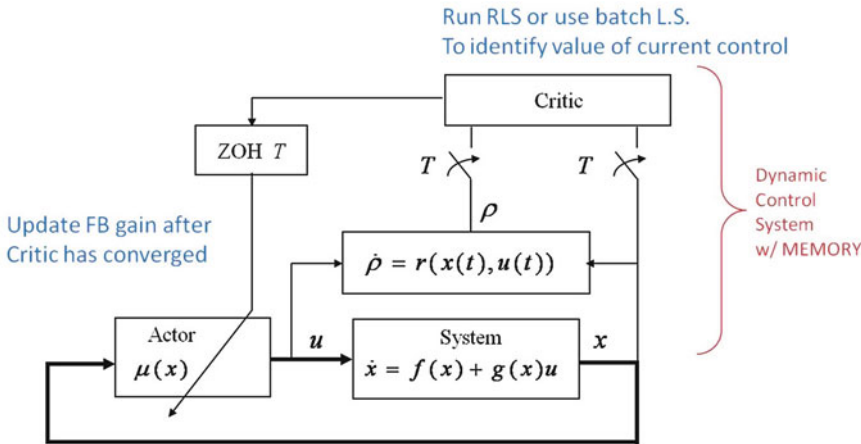
Neural Control and Approximate Dynamic Programming, Fig. 4 Hybrid optimal adaptive controller based on IRL

The work in Vamvoudakis and Lewis (2010) presents a way of finding the optimal control solution in a synchronous manner along with stability and convergence guarantees but with known dynamics. This procedure is more nearly in line with accepted practice in adaptive control.

A synchronous online learning algorithm that avoids the knowledge of drift dynamics is proposed in Vamvoudakis et al. (2013).

Learning in Games

Reinforcement learning techniques have been applied to design adaptive controllers that converge to the solution of two-player zero-sum games in Vamvoudakis and Lewis (2012) and Vrabie et al. (2012), of multiplayer nonzero-sum games in Vamvoudakis et al. (2012a), and of Stackelberg games in Vamvoudakis et al. (2012b). In these cases, the adaptive control structure has multiple loops, with action networks and critic networks for each player. The adaptive controller for zero-sum games finds the solution to the H-infinity control problem online in real time. This adaptive controller does not require any systems dynamics information.

Summary and Future Directions

This entry discusses some neuro-inspired adaptive control techniques. These controllers have multi-loop, multi-timescale structures and can learn the solutions to Hamilton-Jacobi design equations such as the Riccati equation online without knowing the full dynamical model of the system. A method known as Q learning allows the learning of optimal control solutions online, in the discrete-time case, for completely unknown systems. Q learning has not yet been fully investigated for continuous-time systems.

Cross-References

▶ Adaptive Control, Overview
▶ Stochastic Games and Learning
▶ Optimal Control and the Dynamic Programming Principle

Bibliography

Abu-Khalaf M, Lewis FL (2005) Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica 41(5):779–791

Al-Tamimi A, Lewis FL, Abu-Khalaf M (2008) Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. IEEE Trans Syst Man Cybern Part B 38(4):943–949

Lewis FL, Liu D (2012) Reinforcement learning and approximate dynamic programming for feedback control. IEEE Press computational intelligence series. Wiley-Blackwell, Oxford

Lewis FL, Vamvoudakis KG (2011) Reinforcement learning for partially observable dynamic processes: adaptive dynamic programming using measured output data. IEEE Trans Syst Man Cybern Part B 41(1):14–25

Lewis FL, Jagannathan S, Yesildirek A (1999) Neural network control of robot manipulators and nonlinear systems. Taylor and Francis, London

Lewis FL, Campos J, Selmic R (2002) Neuro-fuzzy control of industrial systems with actuator nonlinearities. Society of Industrial and Applied Mathematics Press, Philadelphia

Lewis FL, Vrabie D, Syrmos VL (2012a) Optimal control. Wiley, New York

Lewis FL, Vrabie D, Vamvoudakis KG (2012b) Reinforcement learning and feedback control: using natural decision methods to design optimal adaptive controllers. IEEE Control Syst Mag 32(6):76–105

Slotine JJE, Li W (1987) On the adaptive control of robot manipulators. Int J Robot Res 6(3):49–59

Sutton RS, Barto AG (1998) Reinforcement learning – an introduction. MIT, Cambridge

Vamvoudakis KG, Lewis FL (2010) Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 46(5):878–888

Vamvoudakis KG, Lewis FL (2011) Multi-player non zero sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations. Automatica 47(8):1556–1569

Vamvoudakis KG, Lewis FL (2012) Online solution of nonlinear two-player zero-sum games using synchronous policy iteration. Int J Robust Nonlinear Control 22(13):1460–1483

Vamvoudakis KG, Lewis FL, Hudas GR (2012a) Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality. Automatica 48(8):1598–1611

Vamvoudakis KG, Lewis FL, Johnson M, Dixon WE (2012b) Online learning algorithm for Stackelberg games in problems with hierarchy. In: Proceedings

N

of the 51st IEEE conference on decision and control, Maui pp 1883–1889

Vamvoudakis KG, Vrabie D, Lewis FL (2013) Online adaptive algorithm for optimal control with integral reinforcement learning. Int J Robust Nonlinear Control, Wiley. doi: 10.1002/rnc.3018

Vrabie D, Pastravanu O, Lewis FL, Abu-Khalaf M (2009) Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica 45(2):477–484

Vrabie D, Vamvoudakis KG, Lewis FL (2012) Optimal adaptive control and differential games by reinforcement learning principles. Control engineering series. IET Press, London

Werbos PJ (1989) Neural networks for control and system identification. In: Proceedings of the IEEE conference on decision and control, Tampa

Werbos PJ (1992) Approximate dynamic programming for real-time control and neural modeling. In: White DA, Sofge DA (eds) Handbook of intelligent control. Van Nostrand Reinhold, New York

Nominal Model-Predictive Control

Lars Grüne
Mathematical Institute, University of Bayreuth, Bayreuth, Germany

Abstract

Model-predictive control is a controller design method which synthesizes a sampled data feedback controller from the iterative solution of open-loop optimal control problems. We describe the basic functionality of MPC controllers, their properties regarding feasibility, stability and performance, and the assumptions needed in order to rigorously ensure these properties in a nominal setting.

Keywords

Recursive feasibility; Sampled-data feedback; Stability

Introduction

Model-predictive control (MPC) is a method for the optimization-based control of linear and non-linear dynamical systems. While the literal meaning of "model-predictive control" applies to virtually every model-based controller design method, nowadays the term commonly refers to control methods in which pieces of open-loop optimal control functions or sequences are put together in order to synthesize a sampled data feedback law. As such, it is often used synonymously with "receding horizon control."

The concept of MPC was first presented in Propoǐ (1963) and was reinvented several times already in the 1960s. Due to the lack of sufficiently fast computer hardware, for a while these ideas did not have much of an impact. This changed during the 1970s when MPC was successfully used in chemical process control. At that time, MPC was mainly applied to linear systems with quadratic cost and linear constraints, since for this class of problems algorithms were sufficiently fast for real-time implementation – at least for the typically relatively slow dynamics of process control systems. The 1980s have then seen the development of theory and increasingly sophisticated concepts for linear MPC, while in the 1990s nonlinear MPC (often abbreviated as NMPC) attracted the attention of the MPC community. After the year 2000, several gaps in the analysis of nonlinear MPC without terminal constraints and costs were closed, and increasingly faster algorithms were developed. Together with the progress in hardware, this has considerably broadened the possible applications of both linear and nonlinear MPC.

In this entry, we explain the functionality of nominal MPC along with its most important properties and the assumptions needed to rigorously ensure these properties. We also give some hints on the underlying proofs. The term nominal MPC refers to the assumption that the mismatch between our model and the real plant is sufficiently small to be neglected in the following considerations. If this is not the case, methods from robust MPC must be used (▶ Robust Model-Predictive Control). We describe all concepts for nonlinear discrete time systems, noting that the basic results outlined in this entry are conceptually similar for linear and for continuous-time systems.

Model-Predictive Control

In this entry, we discuss MPC for discrete time control systems of the form

$$x_{\mathbf{u}}(j+1) = f(x_{\mathbf{u}}(j), u(j)), \ x_{\mathbf{u}}(0) = x_0 \quad (1)$$

with state $x_{\mathbf{u}}(j) \in X$, initial condition $x_0 \in \mathbb{X}$, and control input sequence $\mathbf{u} = (u(0), u(1), \ldots)$ with $u(k) \in U$, where the state space X and the control value space U are normed spaces. For control systems in continuous time, one may either apply the discrete time approach to a sampled data model of the system. Alternatively, continuous-time versions of the concepts and results from this entry are available in the literature; see, e.g., Findeisen and Allgöwer (2002) or Mayne et al. (2000).

The core of any MPC scheme is an optimal control problem of the form

$$\text{minimize } J_N(x_0, \mathbf{u}) \quad (2)$$

w.r.t. $\mathbf{u} = (u(0), \ldots, u(N-1))$ with

$$J_N(x_0, \mathbf{u}) : \sum_{j=0}^{N-1} \ell(x_{\mathbf{u}}(j), u(j)) + F(x_{\mathbf{u}}(N)) \quad (3)$$

subject to the constraints

$$u(j) \in \mathbb{U}, x_{\mathbf{u}}(j) \in \mathbb{X} \text{ for } j = 0, \ldots, N-1$$
$$x_{\mathbf{u}}(N) \in \mathbb{X}_0, \quad (4)$$

for control constraint set $\mathbb{U} \subseteq U$, state constraint set $\mathbb{X} \subseteq X$, and terminal constraint set $\mathbb{X}_0 \subseteq X$. The function $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ is called stage cost or running cost; the function $F : \mathbb{X} \to \mathbb{R}$ is referred to as terminal cost. We assume that for each initial value $x_0 \in \mathbb{X}$, the optimal control problem (2) has a solution and denote the corresponding minimizing control sequence by $\mathbf{u}^*$. Algorithms for computing $\mathbf{u}^*$ are discussed in ▶ Optimization Algorithms for Model Predictive Control and ▶ Explicit Model Predictive Control.

The key idea of MPC is to compute the values $\mu_N(x)$ of the MPC feedback law μ_N from the open-loop optimal control sequences $\mathbf{u}^*$. To formalize this idea, consider the closed-loop system

$$x_{\mu_N}(k+1) = f\left(x_{\mu_N}(k), \mu_N\left(x_{\mu_N}(k)\right)\right). \quad (5)$$

In order to evaluate μ_N along the closed-loop solution, given an initial value $x_{\mu_N}(0) \in \mathbb{X}$, we iteratively perform the following steps.

Basic MPC Loop
1. Set $k := 0$.
2. Solve (2)–(4) for $x_0 = x_{\mu_N}(k)$; denote the optimal control sequence by $\mathbf{u}^* = (u^*(0), \ldots, u^*(N-1))$.
3. Set $\mu_N(x_{\mu_N}(k)) : u^*(0)$, compute $x_{\mu_N}(k+1)$ according to (5), set $k := k+1$. and go to (1).

Due to its ability to handle constraints and possibly nonlinear dynamics, MPC has become one of the most popular modern control methods in the industry (▶ Model-Predictive Control in Practice). While in the literature various variants of this basic scheme are discussed, here we restrict ourselves to this most widely used basic MPC scheme.

When analyzing an MPC scheme, three properties are important:
- Recursive Feasibility, i.e., the property that the constraints (4) can be satisfied in Step (ii) in each sampling instant
- Stability, i.e., in particular convergence of the closed-loop solutions $x_{\mu_N}(k)$ to a desired equilibrium x_* as $k \to \infty$
- Performance, i.e., appropriate quantitative properties of $x_{\mu_N}(k)$

Here we discuss these three issues for two widely used MPC variants:
1. MPC with terminal constraints and costs
2. MPC with neither terminal constraints nor costs

In (a), F and $\mathbb{X}_0$ in (3) and (4) are specifically designed in order to guarantee proper performance of the closed loop. In (b), we set $F \equiv 0$ and $\mathbb{X}_0 = \mathbb{X}$. Thus, the choice of ℓ and N in (3) is the most important part of the design procedure.

N

Recursive Feasibility

Since the ability to handle constraints is one of the key features of MPC, it is important to ensure that the constraints $x_{\mu_N}(k) \in \mathbb{X}$ and $\mu_N(x_{\mu_N}(k)) \in \mathbb{U}$ are satisfied for all $k \geq 0$. However, beyond constraint satisfaction, the stronger property $x_{\mu_N}(k) \in \mathbb{X}_N$ is required, where $\mathbb{X}_N$ denotes the *feasible set* for horizon N,

$$\mathbb{X}_N := \{x \in \mathbb{X} | \text{ there exists } \mathbf{u} \text{ such that (4) holds}\}.$$

The property $x \in \mathbb{X}_N$ is called *feasibility* of x. Feasibility of $x = x_{\mu_N}(k)$ is a prerequisite for the MPC feedback μ_N being well defined, because the nonexistence of such an admissible control sequence $\mathbf{u}$ would imply that solving (2) under the constraints (4) in Step (ii) of the MPC iteration is impossible.

Since for $k \geq 0$ the state $x_{\mu_N}(k + 1) = f(x_{\mu_N}(k), u^*(0))$ is determined by the solution of the previous optimal control problem, the usual way to address this problem is via the notion of *recursive feasibility*. This property demands the existence of a set $A \subseteq \mathbb{X}$ such that:

- For each $x_0 \in A$, the problem (2)–(4) is feasible.
- For each $x_0 \in A$ and the optimal control $u^\star$ from (2) to (4), the relation $f(x_0, u^*(0)) \in A$ holds.

It is not too difficult to see that this property implies $x_{\mu_N}(k) \in A$ for all $k \geq 1$ if $x_{\mu_N}(0) \in A$.

For terminal-constrained problems, recursive feasibility is usually established by demanding that the terminal constraint set $\mathbb{X}_0$ is *viable* or *controlled forward invariant*. This means that for each $x \in \mathbb{X}_0$, there exists $u \in \mathbb{U}$ with $f(x, u) \in \mathbb{X}_0$. Under this assumption, it is quite straightforward to prove that the feasible set $A = \mathbb{X}_N$ is also recursively feasible (Grüne and Pannek 2011, Lemma 5.11). Note that viability of $\mathbb{X}_0$ is immediate if $\mathbb{X}_0 = \{x_*\}$ and $x_* \in \mathbb{X}$ is an equilibrium, i.e., a point for which there exists $u_* \in \mathbb{U}$ with $f(x_*, u_*) = x_*$. This setting is referred to as *equilibrium terminal constraint*.

For MPC without terminal constraints, the most straightforward way to ensure recursive feasibility is to assume that the state constraint set $\mathbb{X}$

is viable (Grüne and Pannek 2011, Theorem 3.5). However, checking viability and even more constructing a viable state constraint set is in general a very difficult task. Hence, other methods for establishing recursive feasibility are needed. One method is to assume that the sequence of feasible sets $\mathbb{X}_N$, $N \in \mathbb{N}$ becomes *stationary* for some N_0, i.e., that $\mathbb{X}_{N+1} = \mathbb{X}_N$ holds for all $N \geq N_0$. Under this assumption, recursive feasibility of $\mathbb{X}_{N_0}$ follows, see Kerrigan (2000, Theorem 5.3). However, like viability, stationarity is difficult to verify.

For this reason, a conceptually different approach to ensure recursive feasibility was presented in Grüne and Pannek (2011, Theorem 8.20); a similar approach for linear systems can be found in Primbs and Nevistić (2000). The approach is suitable for stabilizing MPC problems in which the stage cost ℓ penalizes the distance to a desired equilibrium x_* (cf. section "Stability"). Assuming the existence – but not the knowledge – of a viable neighborhood $\mathcal{N}$ of x_*, one can show that any initial point x_0 for which the corresponding open-loop optimal solution satisfies $x_{\mathbf{u}^*}(j) \in \mathcal{N}$ or some $j \leq N$ is contained in a recursively feasible set. The fact that ℓ penalizes the distance to x_* then implies $x_{\mathbf{u}^*}(j) \in \mathcal{N}$ for suitable initial values. Together, these properties yield the existence of recursively feasible sets A_N which become arbitrarily large as N increases.

Stability

Stability in the sense of this entry refers to the fact that a prespecified equilibrium $x_* \in \mathbb{X}$ – typically a desired operating point – is asymptotically stable for the MPC closed loop for all initial values in some set $\mathcal{S}$. This means that the solutions $x_{\mu_N}(k)$ starting in $\mathcal{S}$ converge to x_* as $k \to \infty$ and that solutions starting close to x_* remain close to x_* for all $k \geq 0$. Note that this setting can be extended to time-varying reference solutions; see ▶ Tracking Model Predictive Control.

In order to enforce this property, we assume that the stage cost ℓ penalizes the distance to the equilibrium x_* in the following sense: ℓ satisfies

$$\ell(x_*, u_*) = 0 \text{ and } \alpha_1(|x|) \leq \ell(x, u) \qquad (6)$$

for all $x \in \mathbb{X}$ and $u \in \mathbb{U}$. Here α_1 is a $\mathcal{K}_\infty$ function, i.e., a continuous function $\alpha_1 : [0, \infty) \to [0, \infty)$ which is strictly increasing, unbounded, and satisfies $\alpha_1(0) = 0$. With $|x|$, we denote the norm on X. In this entry, we exclusively discuss stage costs ℓ satisfying (6). More general settings using appropriate detectability conditions are discussed in Rawlings and Mayne (2009, Sect. 2.7) or Grimm et al. (2005) in the context of stabilizing MPC. Even more general ℓ are allowed in the context of economic MPC; see the ▶ Economic Model Predictive Control article.

In case of terminal constraints and terminal costs, a compatibility condition between ℓ and F is needed on $\mathbb{X}_0$ in order to ensure stability. More precisely, we demand that for each $x \in \mathbb{X}_0$ there exists a control value $u \in \mathbb{U}$ such that $f(x, u) \in \mathbb{X}_0$ and

$$F(f(x, u)) - F(x) \leq -\ell(x, u) \qquad (7)$$

holds. Observe that the condition $f(x, u) \in \mathbb{X}_0$ is again the viability condition which we already imposed for ensuring recursive feasibility. Note that (7) is trivially satisfied for $F \equiv 0$ in case of $\mathbb{X}_0 = \{x_*\}$ by choosing $u = u_*$.

Stability is now concluded by using the optimal value function

$$V_N(x_0) := \inf_{\mathbf{u} \text{ s.t. } (4)} J_N(x_0, \mathbf{u})$$

as a Lyapunov function. This will yield stability on $\mathcal{S} = \mathbb{X}_N$, as $\mathbb{X}_N$ is exactly the set on which V_N is defined. In order to prove that V_N is a Lyapunov function, we need to check that V_N is bounded from below and above by $\mathcal{K}_\infty$ functions α_1 and α_2 and that V_N is strictly decaying along the closed-loop solution.

The first amounts to checking

$$\alpha_1(|x|) \leq V_N(x) \leq \alpha_2(|x|) \qquad (8)$$

for all $x \in \mathbb{X}_N$. The lower bound follows immediately from (6) (with the same α_1), and the upper bound can be ensured by conditions

on the problem data (see, e.g., Rawlings and Mayne 2009, Sect. 4.5; Grüne and Pannek 2011, Sect. 5.3).

For ensuring that V_N is strictly decreasing along the closed-loop solutions, we need to prove

$$V_N(f(x, \mu_N(x))) \leq V_N(x) - \ell(x, \mu_N(x)). \quad (9)$$

In order to prove this inequality, one uses on the one hand the dynamic programming principle stating that

$$V_{N-1}(f(x, \mu_N(x))) = V_N(x) - \ell(x, \mu_N(x)). \tag{10}$$

On the other hand, one shows that (7) implies

$$V_{N-1}(x) \geq V_N(x) \qquad (11)$$

for all $x \in \mathbb{X}_N$. Inserting (11) with $f(x, \mu_N(x))$ in place of x into (10) then immediately yields (9). Details of this proof can be found, e.g., in Mayne et al. (2000), Rawlings and Mayne (2009), or Grüne and Pannek (2011). The survey Mayne et al. (2000) is probably the first paper which develops the conditions needed for this proof in a systematic way; a continuous-time version of these results can be found in Fontes (2001).

Summarizing, for MPC with terminal constraints and costs, under the conditions (6)–(8), we obtain asymptotic stability of x_* on $\mathcal{S} = \mathbb{X}_N$.

For MPC without terminal constraints and costs, i.e., with $\mathbb{X}_0 = \mathbb{X}$ and $F \equiv 0$, these conditions can never be satisfied, as (7) will immediately imply $\ell(x, u) = 0$ for all $x \in \mathbb{X}$, contradicting (6). Moreover, without terminal constraints and costs, one cannot expect (9) to be true. This is because without terminal constraints, the inequality $V_{N-1}(x) \leq V_N(x)$ holds, which together with the dynamic programming principle implies that if (9) holds, then it holds with equality. This, however, would imply that μ_N is the infinite horizon optimal feedback law, which – though not impossible – is very unlikely to hold.

Thus, we need to relax (9). In order to do so, instead of (9), we assume the relaxed inequality

$$V_N(f(x, \mu_N(x))) \leq V_N(x) - \alpha \ell(x, \mu_N(x))$$
$$\tag{12}$$

for some $\alpha > 0$ and all $x \in \mathbb{X}$, which is still enough to conclude asymptotic stability of x_* if (6) and (8) hold. The existence of such an α can be concluded from bounds on the optimal value function V_N . Assuming the existence of constants $\gamma_K \geq 0$ such that the inequality

$$V_K(x) \leq \gamma_K \min_{u \in \mathbb{U}} \ell(x, u) \tag{13}$$

holds for all $K = 1, \ldots, N$ and $x \in \mathbb{X}$, there are various ways to compute α from $\gamma_1, \ldots, \gamma_N$, see Grüne (2012, Sect. 3). The best possible estimate for α, whose derivation is explained in detail in Grüne and Pannek (2011, Chap. 6), yields

$$\alpha = 1 - \frac{(\gamma_N - 1) \prod\limits_{i=2}^{N} (\gamma_i - 1)}{\prod\limits_{i=2}^{N} \gamma_i - \prod\limits_{i=2}^{N} (\gamma_i - 1)}. \tag{14}$$

Though not immediately obvious, a closer look at this term reveals $\alpha \to 1$ as $N \to \infty$ if the γ_K are bounded. Hence, $\alpha > 0$ for sufficiently large N.

Summarizing the second part of this section, for MPC without terminal constraints and costs, under the conditions (6), (8), and (13), asymptotic stability follows on $\mathcal{S} = \mathbb{X}$ for all optimization horizons N for which $\alpha > 0$ holds in (14). Note that the condition (13) implicitly depends on the choice of ℓ. A judicious choice of ℓ can considerably reduce the size of the horizon N for which $\alpha > 0$ holds, see Grüne and Pannek (2011, Sect. 6.6) and thus the computational effort for solving (2)–(4).

Performance

Performance of MPC controllers can be measured in many different ways. As the MPC controller is derived from successive solutions of (2), a natural quantitative way to measure its performance is to evaluate the infinite horizon functional corresponding to (3) along the closed loop, i.e.,

$$J_\infty^{c\ell}(x_0, \mu_N) := \sum_{k=0}^{\infty} \ell \left(x_{\mu_N}(k), \mu_N \left(x_{\mu_N}(k) \right) \right)$$

with $x_{\mu_N}(0) = x_0$. This value can then be compared with the optimal infinite horizon value

$$V_\infty(x_0) := \inf_{\mathbf{u}: u(k) \in \mathbb{U}, x_{\mathbf{u}}(k) \in \mathbb{X}} J_\infty(x_0, \mathbf{u})$$

where

$$J_\infty(x_0, \mathbf{u}) := \sum_{k=0}^{\infty} \ell(x_{\mathbf{u}}(k), u(k)).$$

To this end, for MPC with terminal constraints and costs, by induction over (9) and using non-negativity of ℓ, it is fairly easy to conclude the inequality

$$J_\infty^{c\ell}(x_0, \mu_N) \leq V_N(x_0)$$

for all $x \in \mathbb{X}_N$. However, due to the conditions on the terminal cost in (7), V_N may be considerably larger than V_∞ and an estimate relating these two functions is in general not easy to derive (Grüne and Pannek 2011, Examples 5.18 and 5.19). However, it is possible to show that under the same assumptions guaranteeing stability, the convergence

$$V_N(x) \to V_\infty(x)$$

holds for $N \to \infty$ (Grüne and Pannek 2011, Theorem 5.21). Hence, we recover approximately optimal infinite horizon performance for sufficiently large horizon N.

For MPC without terminal constraints and costs, the inequality $V_N(x_0) \leq V_\infty(x_0)$ is immediate; however, (9) will typically not hold. As a remedy, we can use (12) in order to derive an estimate. Using induction over (12), we arrive at the estimate

$$J_\infty^{c\ell}(x_0, \mu_N) \leq V_N(x_0)/\alpha \leq V_\infty(x_0)/\alpha.$$

Since $\alpha \to 1$ as $N \to \infty$, also in this case we obtain approximately optimal infinite horizon performance for sufficiently large horizon N.

Summary and Future Directions

MPC is a controller design method which uses the iterative solution of open-loop optimal control problems in order to synthesize a sampled data feedback controller μ_N. The advantages of MPC are its ability to handle constraints, the rigorously provable stability properties of the closed loop, and its approximate optimality properties. Assumptions needed in order to rigorously ensure these properties together with the corresponding mathematical arguments have been outlined in this entry, both for MPC with terminal constraints and costs and without. Among the disadvantages of MPC are the computational effort and the fact that the resulting feedback is a full state feedback, thus necessitating the use of a state estimator to reconstruct the state from output data (▶ Moving Horizon Estimation).

Future directions include the application of MPC to more general problems than set point stabilization or tracking, the development of efficient algorithms for large-scale problems including those originating from discretized infinite-dimensional control problems, and the understanding of the opportunities and limitations of MPC in increasingly complex environments; see also ▶ Distributed Model Predictive Control.

Cross-References

- ▶ Distributed Model Predictive Control
- ▶ Economic Model Predictive Control
- ▶ Explicit Model Predictive Control
- ▶ Model-Predictive Control in Practice
- ▶ Moving Horizon Estimation
- ▶ Optimization Algorithms for Model Predictive Control
- ▶ Robust Model-Predictive Control
- ▶ Stochastic Model Predictive Control
- ▶ Tracking Model Predictive Control

Recommended Reading

MPC in the form known today was first described in Propoǐ (1963) and is now covered in several monographs, two recent ones being Rawlings and Mayne (2009) and Grüne and Pannek (2011). More information on continuous-time MPC can be found in the survey by Findeisen and Allgöwer (2002). The nowadays standard framework for stability and feasibility of MPC with stabilizing terminal constraints is presented in Mayne et al. (2000); for a continuous-time version, see Fontes (2001). Stability of MPC without terminal constraints was proved in Grimm et al. (2005) under very general conditions; for a comparison of various such results, see Grüne (2012). Feasibility without terminal constraints is discussed in Kerrigan (2000) and Primbs and Nevistić (2000).

Bibliography

Findeisen R, Allgöwer F (2002) An introduction to nonlinear model predictive control. In: 21st Benelux meeting on systems and control, Veldhoven, The Netherlands (see also http://www.tue.nl/en/publication/ep/p/d/ep-uid/252788/), pp 119–141

Fontes FACC (2001) A general framework to design stabilizing nonlinear model predictive controllers. Syst Control Lett 42:127–143

Grimm G, Messina MJ, Tuna SE, Teel AR (2005) Model predictive control: for want of a local control Lyapunov function, all is not lost. IEEE Trans Autom Control 50(5):546–558

Grüne L (2012) NMPC without terminal constraints. In: Proceedings of the IFAC conference on nonlinear model predictive control – NMPC'12, pp 1–13

Grüne L, Pannek J (2011) Nonlinear model predictive control: theory and algorithms. Springer, London

Kerrigan EC (2000) Robust constraint satisfaction: invariant sets and predictive control. PhD thesis, University of Cambridge

Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: stability and optimality. Automatica 36:789–814

Primbs JA, Nevistić V (2000) Feasibility and stability of constrained finite receding horizon control. Automatica 36(7):965–971

Propoǐ A (1963) Application of linear programming methods for the synthesis of automatic sampled-data systems. Avtomat i Telemeh 24:912–920

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison

N

Nonlinear Adaptive Control

A. Astolfi
Department of Electrical and Electronic
Engineering, Imperial College London,
London, UK
Dipartimento di Ingegneria Civile e Ingegneria
Informatica, Università di Roma Tor Vergata,
Roma, Italy

Abstract

We consider the control of nonlinear systems in which parameters are uncertain and may vary. For such systems the control must adapt to the parameter change to deliver closed-loop performance, such as asymptotic stability or tracking. A concise description of available methods and basic adaptive stabilization results, which can be used as building blocks for complex adaptive control problems, are discussed.

Keywords

Adaptive stabilization; Linear parameterization; Lyapunov function; Nonlinear parameterization; Output feedback

Introduction

The adaptive control problem, namely, the problem of designing a feedback controller which contains an *adaptation mechanism* to counteract changes in the parameters of the system to be controlled, is of significant importance in applications. In almost all systems, physical parameters are subject to changes. These may be triggered, for example, by changes in temperature (the volume of a liquid/gas), aging (the friction coefficient of a mechanical system), or normal operation (the mass of the fuel of an aircraft changes during flight, the center of mass of a vehicle is affected by its load).

While adaptive control is naturally associated with the notion of estimation, i.e., the parameters of a system have to be identified to design a controller, it may be possible to design adaptive controllers which do not rely on a *complete* parameter estimation: it is sufficient to estimate the *effect* of the parameters on the control signal.

Adaptive control is different from robust control. In the simplest possible occurrence, the aim of robust control is to design a control law guaranteeing performance specifications for a given range of parameter values. Robust control thus requires some a priori information on the parameter. Adaptive control does not require any a priori information on the parameter, although any such information can be exploited in the controller design, but requires a parameterized model: a model which contains information on the way the parameters affect the dynamics of the system.

The adaptive control problem for general nonlinear systems can be formulated as follows. Consider a nonlinear system described by equations of the form

$$\dot{x} = F(x, u, \theta), \qquad\qquad y = H(x, \theta), \quad (1)$$

where $x(t) \in \mathbb{R}^n$ denotes the state of the system, $u(t) \in \mathbb{R}^m$ denotes the input of the system, $\theta \in \mathbb{R}^q$ denotes the constant unknown parameter, $y(t) \in \mathbb{R}^p$ denotes the measured output, and $F : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n$ and $H : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^p$ are smooth mappings. While we focus on continuous-time systems, similar considerations apply to discrete-time systems. In what follows, for simplicity, we mostly assume that $y = x$: the whole state of the system is available for control design.

The adaptive control problem consists in finding, if possible, a dynamic control law described by equations of the form

$$\dot{\hat{\theta}} = w(x, \hat{\theta}, r), \qquad\qquad (2)$$

$$u = v(x, \hat{\theta}, r), \qquad\qquad (3)$$

with $r(t) \in \mathbb{R}^s$ an exogenous (reference) signal and $w : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^s \to \mathbb{R}^q$ and $v : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^s \to \mathbb{R}^m$ smooth mappings, such

that the closed-loop system, described by the equations

$$\dot{x} = F(x, v(x, \hat{\theta}, r), \theta), \qquad \dot{\hat{\theta}} = w(x, \hat{\theta}, r), \tag{4}$$

has specific properties. For example, one could require that all trajectories be bounded and the x-component of the state converge to a given value $x^{\star}$ (this is the so-called adaptive regulation requirement) or that the input-output behavior of the system from the input r to some user-defined output signal coincide with a given reference model (this is the so-called model reference adaptive control requirement).

A natural way to characterize design specifications for the adaptive control problem and to facilitate its solution is to assume the existence of a known parameter controller, described by the equation

$$u = v^{\star}(x, \theta, r), \tag{5}$$

such that the nonadaptive closed-loop system $\dot{x} = F(x, v^{\star}(x, \theta, r), \theta)$ satisfies given design specifications. In this perspective, the adaptive control problem boils down to the design of the *update law* (2) and of the *feedback law* (3) such that the behavior of the adaptive closed-loop system *matches* that of the nonadaptive closed-loop system $\dot{x} = F(x, v^{\star}(x, \theta, r), \theta)$.

The above description suggests a design method for the *feedback law*: one could replace θ with $\hat{\theta}$ in Eq. (5). This design is often known as certainty equivalence design and lends itself to the interpretation that $\hat{\theta}$ be an estimate for θ. Naturally, one could also modify the feedback law, replacing θ with $\hat{\theta}$ and adding x-dependant terms: this is often called a redesign. Redesign may be guided by various considerations, for example, it may be based on the use of a specific Lyapunov function (yielding the so-called Lyapunov redesign), or by structural properties of the system, or by robustness constraints.

The interpretation of $\hat{\theta}$ as an estimate for θ leads to two similar approaches for the design of the update law. The former, pursued in the so-called indirect adaptive control, relies on the design of a parameter estimator, for example,

using recursive least-square methods. This approach has its roots in identification theory and has been studied in-depth for linear systems. The latter relies on the observation that the design of an update law is equivalent to the design of a (reduced-order) observer for the extended system

$$\dot{x} = F(x, u, \theta), \qquad \dot{\theta} = 0,$$

with output $y = x$. This approach has its roots in the theory of nonlinear observer design.

The approaches described so far relies on a sort of separation principle: the update law and the feedback law are designed separately. While this approach may be adequate for linear systems, for nonlinear systems it is often necessary to design the update law and the feedback law in one step, i.e., the selection of the feedback law depends upon the selection of the update law and vice versa. To illustrate this design method, and provide some explicit adaptive control design tools, we focus on a special class of nonlinear systems: systems which are linearly parameterized in the unknown parameter.

Linearly Parameterized Systems

Consider the system (1) and assume the mapping F is affine in the parameter θ and in the control u, namely,

$$F(x, u, \theta) = f_0(x) + g(x)u + f_1(x)\theta, \tag{6}$$

with $f_0 : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^m$ and $f_1 : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^q$ smooth mappings. For this class of systems, under additional assumptions, it is possible to provide systematic adaptive control design tools. We provide two formal results: additional results (depending on the specific assumptions imposed on the system) may be derived. In both cases the focus is on the adaptive stabilization problem: the goal of the adaptive controller is to render a given equilibrium stable, in the sense of Lyapunov, and to guarantee convergence of the x-component of the state (recall that the state of the adaptive closed-loop system is the vector $(x, \hat{\theta})$).

Theorem 1 *Consider the system (6) and a point x^*. Assume there exist a known parameter controller*

$$u = v_0(x) + v_1(x)\theta,$$

with $v_0 : \mathbb{R}^n \to \mathbb{R}^m$ and $v_1 : \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^q$ smooth mappings, and a positive definite and radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}$, such that $V(x^\star) = 0$ and

$$\frac{\partial V}{\partial x} f^*(x, \theta) < 0$$

for all $x \neq x^\star$.

Then the update law

$$\dot{\hat{\theta}} = -\left(\frac{\partial V}{\partial x} g(x) v_1(x)\right)^\top$$

and the feedback law

$$u = v_0(x) + v_1(x)\hat{\theta}$$

are such that all trajectories of the closed-loop system are bounded and $\lim_{t \to \infty} x(t) = x^\star$.

Theorem 2 *Consider the system (6) and a point x^*. Assume there exists a known parameter controller $u = v(x, \theta)$ such that the closed-loop system*

$$\dot{x} = f^*(x, \theta),$$

where $f^*(x, \theta) = f_0(x) + f_1(x)\theta + g(x)v(x, \theta)$, has a globally asymptotically stable equilibrium at x^*. Assume, in addition, that there exists a mapping $\beta : \mathbb{R}^n \to \mathbb{R}^q$ such that all trajectories of the system

$$\dot{z} = -\left[\frac{\partial \beta}{\partial x} f_1(x)\right] z,$$
$$\dot{x} = f^*(x) + g(x)\left(v(x, \theta + z) - v(x, \theta)\right)$$
$$(7)$$

are bounded and satisfy

$$\lim_{t \to \infty} \left[g(x(t))\left(v(x(t), \theta + z(t)) - v(x(t), \theta)\right)\right] = 0.$$

Then the update law

$$\dot{\hat{\theta}} = -\frac{\partial \beta}{\partial x}\Big[f_0(x) + f_1(x)\big(\hat{\theta} + \beta(x)\big)$$
$$+ g(x)v(x, \hat{\theta} + \beta(x))\Big] \tag{8}$$

and the feedback law

$$u = v(x, \hat{\theta} + \beta(x))$$

are such that all trajectories of the closed-loop system are bounded and $\lim_{t \to \infty} x(t) = x^\star$.

The stability properties of the adaptive closed-loop system in Theorem 1 can be studied with the Lyapunov function $W(x, \hat{\theta}) = V(x) + \frac{1}{2}\|\hat{\theta} - \theta\|^2$, whereas a Lyapunov analysis for the adaptive closed-loop system of Theorem 2 can be carried out, under additional assumptions, via a Lyapunov function of the form $W(x, \hat{\theta}) = V(x) + \frac{1}{2}\|\hat{\theta} - \theta + \beta(x)\|^2$. This suggests that in Theorem 1 $\hat{\theta}$ plays the role of the estimate of θ, whereas in Theorem 2 such a role is played by $\hat{\theta} + \beta(x)$. Note that in none of the theorems, the parameter estimate is required to converge to the true value of the parameters, although in Theorem 2 the feedback law is required to converge, along trajectories, to the known parameter controller. This has a very important, possibly counterintuitive, consequence: the *asymptotic* nonadaptive controller $u = v(x, \hat{\theta}_\infty)$, where $\hat{\theta}_\infty = \lim_{t \to \infty} \hat{\theta}(t)$, provided the limit exists, is not in general a stabilizing controller for system (6).

Example 1 Consider the nonlinear system described by the equation $\dot{x} = u + \theta x^2$, with $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, and $\theta \in \mathbb{R}$. A known parameter controller satisfying the assumptions of Theorem 1 (with and $V(x) = x^2/2$) and of Theorem 2 (with $\beta(x) = x$) is $u = -x - \theta x^2$. The resulting update laws and feedback laws are

$$\dot{\hat{\theta}}_1 = x^3, \qquad u_1 = -x - \hat{\theta}x^2,$$

and

$$\dot{\hat{\theta}}_2 = x, \qquad u_2 = -x - (\hat{\theta} + x)x^2,$$

respectively, the subscripts "1" and "2" are used to refer to the construction in Theorem 1 and 2, respectively.

The basic building blocks in Theorems 1 and 2 can be exploited repeatedly to design adaptive controllers for systems with a specific structure, for example, for systems described by the equations

$$
\begin{aligned}
\dot{x}_1 &= x_2 + \varphi_1(x_1)^\top \theta, \\
\dot{x}_2 &= x_3 + \varphi_2(x_1, x_2)^\top \theta, \\
&\vdots \\
\dot{x}_i &= x_{i+1} + \varphi_i(x_1, \ldots, x_i)^\top \theta, \\
&\vdots \\
\dot{x}_n &= u + \varphi_n(x_1, \ldots, x_n)^\top \theta,
\end{aligned}
\tag{9}
$$

with $x_i(t) \in \mathbb{R}$, for $i = 1, \ldots, n$, $u(t) \in \mathbb{R}$, $\varphi_i : \mathbb{R}^i \to \mathbb{R}^q$, for $i = 1, \ldots, n$, smooth mappings, and $\theta \in \mathbb{R}^q$. Note that the last of the equations (9) can be replaced by

$$
\dot{x}_n = \bar{\theta} u + \varphi_n(x_1, \ldots, x_n)^\top \theta,
$$

with $\bar{\theta} \in \mathbb{R}$, provided its sign is known (this condition may be removed using the so-called Nussbaum gain). The parameter $\bar{\theta}$ is often referred to as the high-frequency gain of the system: a terminology borrowed from linear systems theory.

Output Feedback Adaptive Control

A key feature of the parameterized systems described so far is that these are linearly parameterized in θ. The linear parameterization allows to develop systematic design tools, such as those given in Theorems 1 and 2. Such results, however, require full information on the state of the system. If only partial information on the state is available, one has to combine an estimator of the state with an update law. Such a combination requires either strong assumptions on the system or very specific structures. For example, it is feasible if the system is not only linearly parameterized in the parameter θ, but it is also linearly parameterized in the unmeasured states, namely, it is described by equations of the form

$$
\begin{aligned}
\dot{x}_1 &= x_2 + \psi_1(x_1) + \varphi_1(x_1)^\top \theta, \\
\dot{x}_2 &= x_3 + \psi_2(x_1) + \varphi_2(x_1)^\top \theta, \\
&\vdots \\
\dot{x}_i &= x_{i+1} + \psi_i(x_1) + \varphi_i(x_1)^\top \theta + b_i u, \\
&\vdots \\
\dot{x}_{n-1} &= x_n + \psi_{n-1}(x_1) + \varphi_{n-1}(x_1)^\top \theta + b_{n-1} u, \\
\dot{x}_n &= \psi_n(x_1) + \varphi_n(x_1)^\top \theta + b_n u, \\
y &= x_1
\end{aligned}
$$

with $x_i(t) \in \mathbb{R}$, for $i = 1, \ldots, n$, $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$, $\varphi_i : \mathbb{R} \to \mathbb{R}^q$, and $\psi_i : \mathbb{R} \to \mathbb{R}$, for $i = 1, \ldots, n$, smooth mappings, $\theta \in \mathbb{R}^q$, and $b = [b_i, \cdots, b_{n-1}, b_n]^\top$ unknown, but such that the sign of b_n is known and the polynomial $b_n s^{n-i} + b_{n-1} s^{n-i-1} + \cdots + b_i$ has all roots with negative real part (this implies that the system, with input u and output y, is minimum phase).

Nonlinear Parameterized Systems

Adaptive control of nonlinearly parameterized systems is an open area of research. The design of adaptive controllers relies often upon structural assumptions, for example, the existence of a monotonic parameterization, as in the system described by the equation

$$
\dot{x} = F(x, u) + \Phi(x, \theta),
$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $\theta \in \mathbb{R}^q$, and $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ and $\Phi : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$ smooth mappings and such that, for all x, the mapping Φ satisfies the monotonicity condition

$$
(\theta_a - \theta_b)^\top (\Phi(x, \theta_a) - \Phi(x, \theta_b)) > 0,
$$

for all $\theta_a \neq \theta_b$. Alternatively, the design may exploit the so-called over-parameterization, for example, the equation of the system

$$
\dot{x} = u + \psi_1(x) \sin \theta + \psi_2(x) \cos \theta,
$$

with $x(t) \in \mathbb{R}$, $u(t) \in \mathbb{R}$, and $\theta \in \mathbb{R}$, may be rewritten in over-parameterized form as

$$\dot{x} = u + \psi_1(x)\theta_1 + \psi_2(x)\theta_2,$$

with $\theta_i \in \mathbb{R}$, for $i = 1, 2$. Note that the over-parameterized form *overlooks* the important information that $\theta_1^2 + \theta_2^2 = 1$.

Summary and Future Directions

The problem of adaptive stabilization for nonlinear systems has been discussed. Two conceptual building blocks for the design of stabilizing adaptive controllers have been discussed, and classes of systems for which these blocks allow to explicitly design adaptive controllers have been given. The role of parameter convergence, or lack thereof, has been briefly discussed together with connections between adaptive and observer designs. The difficulties associated with non-full state measurement and with nonlinear parameterization have been also briefly highlighted. Several problems have not been discussed, for example, model reference adaptive control, robust adaptive control, universal adaptive controllers, and the use of projections to incorporate prior knowledge on the parameter. Details on these can be found in the bibliography below.

Cross-References

▶ Adaptive Control, Overview
▶ History of Adaptive Control
▶ Stochastic Adaptive Control
▶ Switching Adaptive Control

Bibliography

Astolfi A, Karagiannis D, Ortega R (2008) Nonlinear and adaptive control with applications. Springer, London
Hovakimyan N, Cao C (2010) L_1 adaptive control theory. SIAM, Philadelphia
Ilchmann A (1997) Universal adaptive stabilization of nonlinear systems. Dyn Control 7(3): 199–213
Ilchmann A, Ryan EP (1994) Universal λ-tracking for nonlinearly-perturbed systems in the presence of noise. Automatica 30(2):337–346
Jiang Z-P, Praly L (1998) Design of robust adaptive controllers for nonlinear systems with dynamic uncertainties. Automatica 34(7):825–840
Kanellakopoulos I, Kokotović PV, Morse AS (1991) Systematic design of adaptive controllers for feedback linearizable systems. IEEE Trans Autom Control 36(11):1241–1253
Krstić M, Kanellakopoulos I, Kokotović P (1995) Nonlinear and adaptive control design. Wiley, New York
Marino R, Tomei P (1995) Nonlinear control design: geometric, adaptive and robust. Prentice-Hall, London
Nussbaum RD Some remarks on a conjecture in parameter adaptive control. Syst Control Lett 3(5):243–246 (1982)
Pomet J-B, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Trans Autom Control 37(6):729–740
Sastry SS, Isidori A (1989) Adaptive control of linearizable systems. IEEE Trans Autom Control 34(11):1123–1131
Spooner JT, Maggiore M, Ordonez R, Passino KM (2002) Stable adaptive control and estimation for nonlinear systems. Wiley, New York
Townley S (1999) An example of a globally stabilizing adaptive controller with a generically destabilizing parameter estimate. IEEE Trans Autom Control 44(11):2238–2241

Nonlinear Filters

Frederick E. Daum
Raytheon Company, Woburn, MA, USA

Abstract

Nonlinear filters estimate the state of dynamical systems given noisy measurements related to the state vector. In theory, such filters can provide optimal estimation accuracy for nonlinear measurements with nonlinear dynamics and non-Gaussian noise. However, in practice, the actual performance of nonlinear filters is limited by the curse of dimensionality. There are many different types of nonlinear filters, including the extended Kalman filter, the unscented Kalman filter, and particle filters.

Keywords

Bayesian; Computational complexity; Curse of dimensionality; Estimation; Extended Kalman filter; Non-Gaussian; Particle filter; Prediction; Smoothing; Stability; Unscented Kalman filter

Description of Nonlinear Filters

Nonlinear filters are algorithms that estimate the state vector (x) of a nonlinear dynamical system given measurements of nonlinear functions of the state vector corrupted by noise. Such filters also quantify the uncertainty in the resulting estimate of the state vector (e.g., using the error covariance matrix). Some nonlinear filters compute the entire probability density of the state vector conditioned on the set of measurements available, rather than computing a point estimate of the state vector (e.g., conditional mean or maximum likelihood). For some applications the conditional probability density of x is highly non-Gaussian (e.g., strongly multimodal). Even if the measurement noise and the process noise and the initial uncertainty in x are all Gaussian, the conditional density of x can be non-Gaussian, owing to the nonlinearities in the dynamics or measurements. The dynamical systems can evolve in continuous time or discrete time, and the measurements can be made in continuous time or at discrete times. The most popular nonlinear filter in practical applications is the extended Kalman filter (EKF), but there are many other families of nonlinear filters, including particle filters, unscented Kalman filters (UKFs), batch least squares, exact finite-dimensional filters, Gaussian sum filters, cubature Kalman filters, etc. Table 1 summarizes the most popular nonlinear filters. The theory for nonlinear filters is relatively simple (see Ho and Lee 1964), but the crucial practical issue is computational complexity, even today with fast modern inexpensive computers, e.g., graphical processing units (GPUs). See Ristic et al. (2004) for a book which is both accessible to engineers and thorough.

Bayesian Formulation of Filtering Problem

The Bayesian approach to nonlinear filters is by far the most popular formulation of the problem (see Ho and Lee 1964), and it has virtually eliminated all other competing theories, because it is simple, general, systematic, and useful. All ten nonlinear filters listed in Table 1 are Bayesian. The Bayesian approach uses a model of the dynamics of x as well as a model of the measurements. For example, discrete-time dynamics and measurement models are typically of the form

$$x(t_{k+1}) = f(x(t_k), t_k) + w(t_k)$$

$$z(t_k) = h(x(t_k), t_k) + v(t_k)$$

in which $x(t_k)$ is the d-dimensional state vector at time t_k, $z(t_k)$ is the m-dimensional measurement vector at time t_k, v is the measurement noise, and w is the so-called process noise. Both v and w are often modeled as Gaussian zero-mean random processes with statistically independent values at distinct discrete times, but these models could be highly non-Gaussian with statistically correlated random values. The initial probability density of x before any measurements are available is also used in the Bayesian formulations. Real physical systems are most commonly modeled as evolving in continuous time using Itô stochastic differential equations:

$$dx = f(x(t), t)dt + dw$$

However, most engineers would rather think of the above Itô equation as an ordinary differential equation driven by Gaussian white noise:

$$dx/dt = f(x(t), t) + dw/dt$$

Mathematicians prefer the Itô equation to avoid the embarrassment that the time derivative of w(t) does not exist. For details of stochastic calculus, see Jazwinski (1998). Such mathematical subtleties rarely cause any trouble in practical engineering applications. We emphasize, however, that it is important to correctly model continuous-

N

Nonlinear Filters, Table 1　Summary of nonlinear filters

Nonlinear filter	Conditional probability density	Computational complexity	Comments	References
1. Extended Kalman filter (EKF)	Gaussian	d^3	Gives good accuracy for many practical applications but can be highly suboptimal in difficult problems	Gelb et al. (1974)
2. Unscented Kalman filter(UKF)	Gaussian	d^3	Often the UKF beats the EKF, but sometimes the EKF is better than the UKF; see Noushin (2008) for details	Julier and Uhlmann (2003)
3. Batch least squares	Gaussian	d^3	Often beats the EKF accuracy but can fail for multimodal or other strongly non-Gaussian densities	Sorenson (1980)
4. Particle filter	Arbitrary	Varies from d^3 to exponential in d, depending on many features of the problem	Often beats the EKF accuracy but can fail due to the curse of dimensionality and particle degeneracy and ill-conditioning	Doucet (2011)
5. Cubature Kalman filter	Gaussian	d^3	Sometimes beats the EKF and UKF for difficult nonlinear non-Gaussian problems, but not always	Haykin (2010)
6. Gaussian sum	Arbitrary	Varies from d^3 to exponential in d, depending on many features of the problem	Beats the EKF for certain difficult nonlinear non-Gaussian problems	Sorenson (1988)
7. Exact finite-dimensional filters	Exponential family	d^3	Beats the EKF for certain difficult nonlinear non-Gaussian problems	Daum (2005)
8. Implicit particle filters	Arbitrary	Suffers from the curse of dimensionality (i.e., computation time grows exponentially in d)	Only low-dimensional numerical examples have been published so far	Chorin (2009)
9. Particle flow filter	Arbitrary	Faster than standard particle filters by many orders of magnitude for high-dimensional problems (but unfortunately there is no explicit formula for computation time)	Beats the EKF by orders of magnitude for certain difficult nonlinear non-Gaussian problems	Daum (2013)
10. Numerical solution of Fokker-Planck equation	Arbitrary	Suffers from the curse of dimensionality (i.e., computation time grows exponentially in d)	Beats the EKF by orders of magnitude for certain difficult nonlinear non-Gaussian problems	Ristic (2004)

time random processes for the evolution of the state vector (x) in many practical applications. Similarly, one can model measurements in continuous time using Itô calculus:

$$dz = h(x(t), t)dt + dv$$

Most engineers consider continuous-time measurement models as impractical and unnecessarily complicated mathematically, because digital computers always require discrete-time measurements and there are no practical analog computers that can be used for nonlinear filtering, owing to the overwhelming superiority of digital computers in terms of accuracy, stability, dynamic range, and flexibility. Nevertheless, there are many papers published by researchers using continuous-time measurement models. But the vast majority of practical papers on nonlinear filters use discrete time measurement models for obvious reasons. This contrasts sharply with the practical importance of correctly modeling continuous-time random processes for the evolution of the state vector (x).

Nonlinear Filter Algorithms

There is no universally best nonlinear filter for all applications, and there is much debate about which is the best nonlinear filter for any given application. Even if we knew the best nonlinear filter for a given computer, the answer could be very different for a different computer; in particular, some filters can exploit massively parallel processing architectures, whereas others cannot. Research and development of nonlinear filters should continue rapidly for the foreseeable future. More generally, there is no universal theory of computational complexity for practical algorithms of this type; perhaps the closest approximation to such a theory is "information-based complexity" (IBC); e.g., see Traub and Werschulz (1998) and Dick et al. (2013). The estimation accuracy of x and the computational complexity of the nonlinear filter are intimately connected, as shown below for particle filters.

There is no useful way to quantify the computational complexity of nonlinear filters without also quantifying estimation accuracy of x. This contrasts with standard computational complexity theory (e.g., P vs. NP) because we are interested in approximations rather than exact solutions. This is the basic idea of IBC. In practice, engineers compare the estimation accuracy and computational complexity of different nonlinear filters using Monte Carlo simulations for specific applications and specific computers.

The most active area of current research in nonlinear filters is focused on particle filters, which have the promise of optimal accuracy for essentially any nonlinear filter problem, at the cost of very high computational complexity for high-dimensional problems. In the early days (1994–2004), researchers often asserted that particle filters "beat the curse of dimensionality," but it is well known today that this assertion is wrong (e.g., see Daum 2005). Unfortunately, there is no useful theory of computational complexity for particle filters, but rather the currently available theory gives asymptotic bounds on accuracy with generic "constants." Such bounds on the variance of estimation error are generally of the form c/N in which N is the number of particles and c is the generic so-called constant. But we know that the so-called constant actually varies by many orders of magnitude depending on the specifics of the problem, including the following: (1) dimension of the state vector being estimated, (2) uncertainty in the initial state vector, (3) measurement accuracy, (4) stability of the dynamical system that describes the time evolution of the state vector, (5) geometry of the conditional probability densities (e.g., unimodal, log-concave, multimodal, etc.), (6) Lipschitz constants of the log probability densities, (7) curvature of the nonlinear dynamics and measurements, (8) ill-conditioning of the Fisher information matrix for the estimation problem, etc. Moreover, there are no tight bounds on the so-called constant c for practical nonlinear filter problems, but rather the best bounds for simple MCMC problems are known to be 30 orders of magnitude too large; see Dick et al. (2013).

Discrete-Time Measurement Models

Research papers on nonlinear filters are often mathematically abstract, but advanced math is not required for practical engineering applications (e.g., see Ho and Lee 1964). In particular, one can avoid the advanced stochastic mathematics used for continuous-time measurements by using discrete-time measurements, which is the practical case of interest anyway, owing to the use of digital computers to implement such algorithms. The notion that continuous-time measurements results in simpler, better, or more elegant results for nonlinear filters is misleading; for example, we have the elegant innovation theory for continuous-time measurements (Kailath 1970), but this theory is not applicable for discrete-time measurements, likewise with the elegant formula for propagating the conditional mean for continuous-time measurements (the so-called Fujisaki-Kallianpur-Kunita formula). More generally, the simple discrete-time version of Bayes' rule suffices for practical real-world engineering applications; there is rarely a need to employ the more complex continuous-time version. The discrete time formula for Bayes' rule is simply

$$p(x(t_k), t_k|Z_k) = p(x(t_k), t_k|Z_{k-1})p(z_k|x(t_k))/p(z_k|Z_{k-1})$$

in which

$p(x(t_k), t_k|Z_k)$ = probability density of x at time t_k conditioned on Z_k; this is also called the "posteriori probability density"

$x(t)$ = state vector of the dynamical system at time t

Z_k = set of all measurements up to and including time t_k

z_k = measurement vector at time t_k

$p(z_k|x(t_k))$ = probability density of z_k conditioned on $x(t_k)$; this is also called the "likelihood"

$p(A|B)$ = probability density of A conditioned on B

This is all one needs to know about Bayes' rule for practical engineering applications of nonlinear filtering; see Ho and Lee (1964). Bayes' rule is a simple formula that multiplies two probability densities and normalizes it by dividing by $p(z_k|Z_{k-1})$. In most applications, there is no need to normalize the density, and hence, Bayes' rule for the unnormalized conditional density is even simpler:

$$p(x(t_k), t_k|Z_k) = p(x(t_k), t_k|Z_{k-1})p(z_k|x(t_k))$$

We see that Bayes' rule for the unnormalized conditional density is simply a multiplication of two densities (i.e., the likelihood and the prior).

Summary and Future Directions

In practical applications, the most popular nonlinear filter is the extended Kalman filter (EKF), followed by the unscented Kalman filter (UKF). These two filters give good accuracy and robust performance for many practical applications. The computational complexity of both the EKF and UKF grows as the cube of the dimension of the state vector, and hence, they are very practical to run in real time on laptops or PCs for many real- world applications. But there are also many difficult nonlinear or non-Gaussian problems for which the EKF and UKF give suboptimal accuracy, and in some cases, they give surprisingly bad accuracy. The accuracy of optimal nonlinear filters is limited by the curse of dimensionality. We know how to write the equations for the optimal nonlinear filter, but the solution generally takes an exponentially increasing time to compute as the dimension of the state vector grows. There are many different kinds of nonlinear filters, and this is still an active field of research, as shown in Crisan and Rozovskii (2011). Future research is likely to exploit advances in computational complexity theory for approximation of functions in the style of information-based complexity (IBC) rather than P vs. NP theory. This is because we want good fast approximations rather than exact algorithms. A lucid introduction to IBC is Traub and Werschulz (1998), and recent work is surveyed in Dick et al. (2013). Another fruitful direction of research is to exploit the recent advances in transport theory,

as explained in Daum (2013); the best introduction to transport theory is the book by Villani (2003), which is very accessible yet thorough. Research in exact finite-dimensional filters is difficult but could yield substantial improvements in accuracy and computational complexity; for example, see Benes (1981), Marcus (1984), and Daum (2005). Progress in nonlinear filter research could be inspired by many diverse fields, including fluid dynamics, quantum chemistry, quantum field theory, gauge theory, string theory, Lie superalgebras, Lie supergroups, and neuroscience. An important open research topic is the stability of nonlinear filters, which is obviously a fundamental limitation to good theoretical upper bounds on estimation error. We still do not have a practical theory of stability for nonlinear filters. Perhaps the closest approximation to such a theory is the lucid paper by van Handel (2010), which makes an interesting attempt at understanding the stability of nonlinear filters. In particular, van Handel's paper aims to generalize Kalman's theory of stability for the Kalman filter by connecting stability with the essence of controllability and observability. A good survey of what is known about stability theory for nonlinear filters is given in various articles in Crisan and Rozovskii (2011).

Cross-References

▸ Estimation, Survey on
▸ Extended Kalman Filters
▸ Kalman Filters
▸ Particle Filters

Bibliography

Arasaratnam I, Haykin S, Hurd TR (2010) Cubature Kalman filtering for continuous-discrete systems. IEEE Trans Signal Process 58:4977–4993

Benes V (1981) Exact finite-dimensional filters for certain diffusions with nonlinear drift. Stochastics 5:65–92

Chorin A, Tu X (2009) Implicit sampling for particle filters. Proc Natl Acad Sci 106:17249–17254

Crisan D, Rozovskii B (eds) (2011) Oxford handbook of nonlinear filtering. Oxford University Press, Oxford

Daum F (2005) Nonlinear filters: beyond the Kalman filter. IEEE AES Magazine 20:57–69

Daum F, Huang J (2013a) Particle flow with non-zero diffusion for nonlinear filters. In: Proceedings of SPIE conference, San Diego

Daum F, Huang J (2013b) Particle flow and Monge-Kantorovich transport. In: Proceedings of IEEE FUSION conference, Singapore

Dick J, Kuo F, Peters G, Sloan I (eds) (2013) Monte Carlo and quasi-Monte Carlo methods 2012. Proceedings of conference, Sydney. Springer, Heidelberg

Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovskii B (eds) The Oxford handbook of nonlinear filtering. Oxford University Press, Oxford pp 656–704

Gelb A et al (1974) Applied optimal estimation. MIT, Cambridge

Ho Y-C, Lee RCK (1964) A Bayesian approach to problems in stochastic estimation and control. IEEE Trans Autom Control 9:333–339

Jazwinski A (1998) Stochastic processes and filtering theory. Dover, Mineola

Julier S, Uhlmann J (2004) Unscented filtering and nonlinear estimation. IEEE Proc 92:401–422

Kailath T (1970) The innovations approach to detection and estimation theory. Proc IEEE 58: 680–695

Kushner HJ (1964) On the differential equations satisfied by conditional probability densities of Markov processes. SIAM J Control 2:106–119

Marcus SI (1984) Algebraic and geometric methods in nonlinear filtering. SIAM J Control Optim 22: 817–844

Noushin A, Daum F (2008) Some interesting observations regarding the initialization of unscented and extended Kalman filters. In: Proceedings of SPIE conference, Orlando

Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter. Artech House, Boston

Sorenson HW (1974) On the development of practical nonlinear filters. Inf Sci 7:253–270

Sorenson HW (1980) Parameter estimation. Marcel-Dekker, New York

Sorenson HW (1988) Recursive estimation for nonlinear dynamic systems. In: Spall J (ed) Bayesian analysis of time series and dynamic models. Marcel-Dekker, New York, pp 127–165

Stratonovich RL (1960) Conditional Markov processes. Theory Probab Appl 5:156–178

Traub J, Werschulz A (1998) Complexity and information. Cambridge University Press, Cambridge

van Handel R (2010) Nonlinear filters and system theory. In: Proceedings of 19th international symposium on mathematical theory of networks and systems, Budapest

Villani C (2003) Topics in optimal transportation. American Mathematical Society, Providence

Zakai M (1969) On the optimal filtering of diffusion processes. Z fur Wahrscheinlichkeitstheorie und verw Geb 11:230–243

N

Nonlinear Sampled-Data Systems

Dragan Nesic[1] and Romain Postoyan[2,3]
[1]Department of Electrical and Electronic Engineering, The University of Melbourne, Melbourne, VIC, Australia
[2]Université de Lorraine, CRAN, France
[3]CNRS, CRAN, France

Abstract

Sampled-data systems are control systems in which the feedback law is digitally implemented via a computer. They are prevalent nowadays due to the numerous advantages they offer compared to analog control. Nonlinear sampled-data systems arise in this context when either the plant model or the controller is nonlinear. While their linear counterpart is now a mature area, nonlinear sampled-data systems are much harder to deal with and, hence, much less understood. Their inherent complexity leads to a variety of methods for their modeling, analysis, and design. A summary of these methods is presented in this entry.

Keywords

Discrete time; Nonlinear; Sampled data; Sampler; Zero-order hold

Introduction

Definition: A control system in which a continuous-time plant is controlled by a digital computer is referred to as a *sampled-data control system* or simply a *sampled-data system* (Chen and Francis 1994); see Fig. 1. *Nonlinear sampled-data systems* arise when either the model of the plant or the controller is nonlinear; otherwise the system is referred to as a linear sampled-data system.

Motivation: Sampled-data control is preferable to continuous-time (analog) control for a range of reasons including reduced cost, reduced wiring, more robust hardware, easier and more flexible programming, and so on. Nowadays, a large majority of controllers are implemented on digital computers, and, hence, sampled-data systems are prevalent in practice. On the other hand, nonlinear plant models are necessary in numerous applications when a wide range of operating conditions need to be considered or when truly nonlinear phenomena, such as friction or state/input constraints, are not negligible. Hence, there are many situations where nonlinear plant models are essential, such as vertical takeoff and landing of an aircraft, robots, automotive engines, and biochemical reactors, to name a few. It has to be noted that the nonlinearity may also come from the controller even when we consider linear plants as it is the case in adaptive control or model predictive control with constraints, for example.

Structure of sampled-data systems: Figure 1 presents a typical structure of a sampled-data system which consists of a continuous-time plant, an analog-to-digital (A/D) converter (i.e., a sampler), a digital-to-analog (D/A) converter (i.e., a hold device), and a discrete-time controller.

The A/D converter takes measurements $y(t_k)$ of a continuous-time output signal $y(t)$, such as temperature or pressure, at sampling time instants $t_k, k = 0, 1, \ldots$ and sends them to the control algorithm. The measurements are obtained with finite precision (i.e., they are quantized); this effect is not considered in this entry. The sampling instants t_k are often equidistant, that is, $t_k = kT$, $k = 0, 1, \ldots$, where the distance T between any two consecutive sampling instants is referred to as the *sampling period*. The sampling period is an important degree of freedom in the design of sampled-data systems and it needs to be carefully selected.

The control algorithm is discrete in nature. It takes the sequence of measurements $y(t_k)$ and processes them to produce a sequence of control values $u(t_k)$. The D/A converter converts the sequence of control values $u(t_k)$ into a continuous-time signal $u(t)$ that drives the actuators which control the plant. Typically, a zero-order hold is used, i.e., $u(t) = u(t_k), \forall t \in [t_k, t_{k+1})$. However, it is possible to use other types of holds.

Nonlinear Sampled-Data Systems, Fig. 1
Sampled-data (control) system
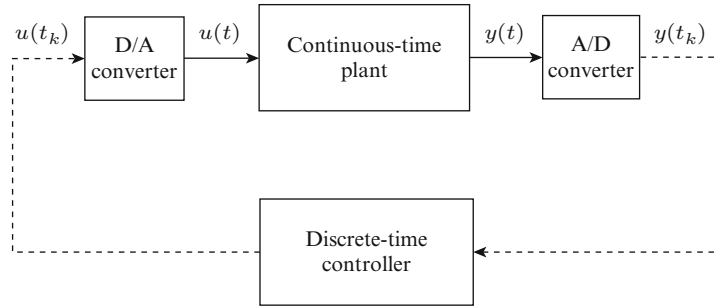
Note that the system in Fig. 1 can be generalized in many ways. An important generalization is *multi-rate sampling* where the output of the system is sampled at one sampling rate while the control inputs are updated at a different sampling rate. Another generalization are *networked control systems* which are discussed in the last section.

Modeling

The combination of continuous-time and discrete-time components renders the analysis and the design of sampled-data systems challenging. Still, linear systems allow for computationally efficient analysis and design techniques that benefit from the z and δ transforms, as well as convex optimization (Chen and Francis 1994). Nonlinear sampled-data systems, one the other hand, are much harder to deal with since the aforementioned methods do not apply in this case. This inherent difficulty has led to a variety of models for different analysis and design methods:

1. Continuous-time models
2. Discrete-time models
3. Sampled-data models

We discuss bellow each of these models, their features, and the analysis or design methods that exploit them.

Continuous-time models basically ignore the sampling process and assume that all signals are continuous time. They are the coarsest approximation of the sampled-data system and they are useful only for very small sampling periods. Nevertheless, they are invaluable and are used as the first step in the controller/observer design in the so-called *emulation* design approach.

Discrete-time models only capture the behavior of the sampled-data system at sampling instants. Indeed, they ignore the inter-sample behavior of the system and this is their main drawback. There are two ways in which nonlinear discrete-time models arise: (i) from the identification of the plant model using the sampled measurements and (ii) from the discretization of a known continuous-time plant model. For instance, black box identification methods often lead to nonlinear discrete-time models in input-output form, such as NARMA (nonlinear autoregressive moving average) models (Chen et al. 1989; Juditsky et al. 1995). Depending on the approximating functions used, the nonlinearities can be polynomial, neural network type, fuzzy type, and so on. On the other hand, the discretization of the continuous-time plant model requires an exact analytic solution of a set of nonlinear differential equations. When such an analytic solution exists, we can obtain the exact discrete-time models of the system; this is typically assumed for linear plants. Nonlinear sampled-data systems are different from their linear counterparts in that it is typically impossible to obtain the exact discrete-time model and only approximate discrete-time models are available for analysis and design (Nešić et al. 1999; Nešić and Teel 2004).

Sampled-data models capture the true behavior of the sampled-data system including its

inter-sample behavior. There are several ways in which this can be achieved. One way is to model the piecewise constant signals that arise from zero-order hold devices as signals with a time-varying delay; this gives rise to time-delay nonlinear models (Teel et al. 1998). Another recently proposed approach is to model nonlinear sampled-data systems as hybrid dynamical systems (Goebel et al. 2012). An extensive analysis and design toolbox has been developed for hybrid dynamical systems and these results can be used for nonlinear sampled-data systems. Another class of models, based on the so-called lifting, has been applied for linear systems where the system is represented as a discrete-time system with infinite dimensional input and output spaces. While this approach has been very successful in the linear context (Chen and Francis 1994), it appears that it is not as useful for nonlinear systems due to difficulties arising from harder analysis and prohibitive computational requirements.

The Main Issues and Analysis

Controllability/observability: Issues arising due to sampling in linear systems transfer to the nonlinear context although they are less understood in this case. For instance, it is well known that sampling may "destroy" the controllability and/or observability properties of the system (Chen and Francis 1994). In other words, if the continuous-time plant model is controllable/observable, then the corresponding exact discrete-time model of the plant may not verify these properties for some sampling periods. A simple test is available for linear systems to avoid this phenomenon, but we are not aware of similar results in the nonlinear context.

Finite escape times: A major difference between continuous-time linear and nonlinear systems is that the former have well defined solutions for constant control inputs and arbitrarily long sampling periods. This is not the case, in general, for nonlinear systems as they may exhibit finite escape times. In other words, for a constant input it may happen for some initial

conditions of a nonlinear system that solutions blow up within a time that is shorter than the sampling period. As a consequence, for such an initial condition and input, the exact discrete-time system cannot be defined. This is a fundamental obstacle to achieving global stability results for nonlinear systems if the sampling period is fixed and independent of the size of the initial state. Nevertheless, it is possible to ensure semi-global stability properties for very general nonlinear systems which means that any compact domain of convergence can be achieved if the sampling period is sufficiently reduced (Nešić and Teel 2004).

Model structure is changed: An important issue for nonlinear sampled-data systems is that the sampling modifies the structure of the model. When the continuous-time plant model has a certain structure, such as triangular or affine in the input, the corresponding exact discrete-time model will not inherit it; see Monaco and Normand-Cyrot (2007) and Yuz and Goodwin (2005). This significantly complicates the design of sampled-data systems via the discrete-time approach since many nonlinear design techniques, like backstepping or forwarding, are heavily reliant on the structure of the model.

Zero dynamics: Probably the most significant aspect of the changed structure are the so-called sampling zeros. In linear systems, it is well known that if a continuous-time linear system of relative degree $r \geq 2$ is sampled, then generically for fast sampling the discrete-time models of the plant will have relative degree $r = 1$. In other words, sampling introduces extra zeros in the model which are often unstable and thus render the system non-minimum phase. It is well known that the controller design is much harder for non-minimum phase systems, and, moreover, there are certain fundamental performance limitations in this case. Recently, results that extend the notion of sampling zeros to the nonlinear sampled-data systems have been reported; see the references in Monaco and Normand-Cyrot (2007).

Passivity: Some plant properties like passivity are much more restrictive in discrete time than in continuous time. Indeed, it is necessary for a

continuous-time plant to have relative degree 1 or 0 to be passive, whereas only relative degree 0 discrete-time plants may possess this property. In other words, an exact discrete-time model of a passive continuous-time plant of relative degree 1 will not be passive; that is, sampling typically destroys passivity.

Controller Design

Linearization: The simplest way to design sampled-data nonlinear systems is to linearize the plant at a given operating point. In this case, the nonlinear plant dynamics are approximated by a linear model around a chosen equilibrium, and then any of the linear sampled-data techniques can be applied to the linearized model. The obtained solution is then implemented on the true nonlinear plant. The drawback of this technique is that the solution would typically perform well only in the vicinity of the selected equilibrium point.

Nonlinear methods: An alternative is to perform designs that rely on a nonlinear plant model. These approaches can be divided into feedback linearization, emulation design method, (approximate and exact) discrete-time design method, and sampled-data design method.

Feedback linearization: Some classical problems, like feedback linearization, are harder for sampled-data systems than continuous-time ones. It was shown that a class of discrete-time nonlinear systems for which feedback linearization is possible is smaller than the corresponding class of continuous-time systems (Grizzle 1987). This has led to approximate feedback linearization techniques which consider achieving feedback linearization approximately with an error that can be reduced by reducing the length of the sampling period (Arapostathis et al. 1989).

Continuous-time design method (Emulation design): Emulation is a design technique consisting of two steps. In the first step, a continuous-time controller or observer is designed for the continuous-time plant while ignoring sampling to achieve appropriate stability, performance, and/or robustness guarantees. In the second step,

the designed controller/observer is discretized for implementation and the sampling period is reduced sufficiently for the method to work. This method is approximate since the continuous-time plant model approximates well the sampled-data systems only for sufficiently small sampling periods. The discretization can be done using various implicit or explicit Runge-Kutta methods, such as the forward or backward Euler method (Monaco and Normand-Cyrot 2007; Yuz and Goodwin 2005). The emulation method is probably the best understood of all design methods. It was shown that a range of stability properties that can be cast in terms of dissipation inequalities are preserved in an appropriate sense under the emulation approach (Laila et al. 2002). Moreover, nonconservative estimates of the upper bound for the required sampling period in emulation have been reported recently (Nešić et al. 2009).

Exact discrete-time design method: Exact discrete-time design method assumes that an exact discrete-time model of the plant is available to the designer; see Kötta (1995) and the references cited therein. This approach is reasonable when black box identification techniques are used for modeling. Moreover, in some rare cases it is possible to obtain the exact discrete-time model of the plant by integrating the continuous-time model with fixed inputs (assuming the zero-order hold is used). This is the case when the plant dynamics are linear while the control law is nonlinear (e.g., adaptive control) or the plant is linear with state/input constraints, which is a setup often used in the model predictive control. The literature on exact discrete-time design method is vast and many of the nonlinear continuous-time design techniques, like backstepping, forwarding, and passivity-based designs, are extended to discrete-time nonlinear systems; see Kötta (1995) and Grizzle (1987). A drawback of these methods is that they assume a special structure of the discrete-time nonlinear model, such as upper or lower triangular structure, which is typically much more restrictive in discrete-time than in continuous-time due to the loss of structure due to sampling that was discussed earlier.

Approximate discrete-time design method: Due to the nonlinearity, it is impossible in most cases to obtain an exact discrete-time plant model by integrating its continuous-time model equations; instead, a range of approximate discrete-time plant models, such as Runge-Kutta, can be used for controller/observer design. It was recently shown that this design method may lead to disastrous consequences where the controller stabilizes the approximate discrete-time plant model for all (arbitrarily small) sampling periods while the same controller destabilizes the exact discrete-time plant model for all sampling periods; see Nešić and Teel (2004) and Nešić et al. (1999). This is true even for linear systems and some commonly used discretization techniques and controller designs. These considerations have led to the development of a framework for controller design based on approximate discrete-time models (Nešić et al. 1999; Nešić and Teel 2004). This framework provides checkable conditions on the continuous-time plant model, the approximate discrete-time model and the controller that guarantee that the controllers designed in this manner would stabilize the exact discrete-time model and, hence, the nonlinear sampled-data system for sufficiently small sampling periods. The design is based on families of approximate discrete-time models parameterized with the sampling period, and the design objectives are more demanding than in the continuous-time nonlinear systems. Ideas from numerical analysis are adapted to this context. This framework was used to design controllers and observers for classes of nonlinear sampled-data systems where typically Euler approximate discretization is employed to generate the approximate discrete-time model.

Sampled-data design method: Both emulation and discrete-time design methods have their drawbacks. Indeed, the former method ignores the sampling at the design stage, whereas the latter method ignores and may produce unacceptable inter-sampling behavior. Thus, methods that use a sampled-data model of the plant for design are much more attractive. There are two possible ways in which this can be achieved for nonlinear sampled-data systems.

The first approach consists of representing nonlinear sampled-data systems as systems with time-varying delays (Teel et al. 1998). However, controller design tools for such systems need to be further developed.

The second approach involves representing the nonlinear sampled-data system as a hybrid dynamical system. Recent advances on modeling and analysis of hybrid dynamical systems (Goebel et al. 2012) offer great opportunities in this context, but the full potential of this approach is still to be exploited. Nonlinear sampled-data systems are just a small subclass of hybrid dynamical systems, and developing specific analysis and design tools tailored to this class of systems seems promising.

It should be emphasized that there are many related techniques, such as discrete-time adaptive control and model predictive control, that deal with classes of nonlinear sampled-data systems but are not a part of the mainstream nonlinear sampled-data literature.

Summary and Future Directions

Summary: Sampled-data control systems are nowadays prevalent and there are many situations where nonlinear models need to be used to deal with wider ranges of operating conditions, more restrictive constraints, and enhanced performance specifications. Despite their increasing importance, the design of nonlinear sampled-data systems remains largely unexplored, and it is much less developed than its continuous-time counterpart. A variety of models, analysis, and design techniques make nonlinear sampled-data literature very diverse and a comprehensive textbook reference or a unifying approach is still missing. Many open questions remain for nonlinear sampled-data systems, such as results on multi-rate sampling, design techniques based on sampled-data models, and other generalizations which are discussed below.

Future Directions: In the 1990s, a new generation of digitally controlled systems has evolved from the more classical sampled-data systems

which are generally referred to as networked control systems (NCS); see Heemels et al. (2010) and the references cited therein. These systems exploit digital wired or wireless communication networks within the control loops. Such a setup is introduced to reduce the cost, weight, and volume of the engineered systems, but its special structure imposes new challenges due to the communication constraints, data packet dropouts, quantization of data, varying sampling periods, time delays, etc. At the same time, these systems provide new flexibilities due to the distributed computation within the control system that can be used to improve the performance and mitigate some of the undesirable network effects on the overall system performance. Moreover, embedded microprocessors allow for event-triggered and self-triggered sampling (Anta and Tabuada 2010) that are still largely unexplored especially for nonlinear systems. Design of NCS was identified as one of the biggest challenges to the control research community in the twenty-first century, and more than a decade of intense research on this topic still has not provided a comprehensive and unifying approach for their analysis and design. Novel results on modeling and Lyapunov stability theory for (nonlinear) hybrid dynamical systems appear to offer the right analysis design tools but they are still to be converted into efficient and easy-to-use design tools in the control engineers' toolbox.

Cross-References

▶ Event-Triggered and Self-Triggered Control
▶ Hybrid Dynamical Systems, Feedback Control of
▶ Optimal Sampled-Data Control
▶ Sampled-Data Systems

Bibliography

Anta A, Tabuada P (2010) To sample or not to sample: self-triggered control for nonlinear systems. IEEE Trans Autom Control 55:2030–2042

Arapostathis A, Jakubczyk B, Lee HG, Marcus SI, Sontag ED (1989) The effect of sampling on linear equivalence and feedback linearization. Syst Control Lett 13:373–381

Chen T, Francis B (1994) Optimal sampled-data systems. Springer, New York

Chen S, Billings SA, Luo W (1989) Orthogonal least squares methods and their application to non-linear system identification. Int J Control 50: 1873–1896

Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems. Princeton University Press, Princeton

Grizzle JW (1987) Feedback linearization of discrete-time systems. Syst Control Lett 9:411–416

Heemels M, Teel AR, van de Wouw N, Nešić D (2010) Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance. IEEE Trans Autom Control 55:1781–1796

Juditsky A, Hjalmarsson H, Benveniste A, Delyon B, Ljung L, Sjöberg J, Zhang Q (1995) Nonlinear black-box models in system identification: mathematical foundations (original research article). Automatica 31:1725–1750

Khalil HK (2004) Performance recovery under output feedback sampled-data stabilization of a class of nonlinear systems. IEEE Trans Autom Control 49:2173–2184

Kötta U (1995) Inversion method in the discrete-time nonlinear control systems synthesis problems. Lecture notes in control and information sciences, vol 205. Springer, Berlin

Laila DS, Nešić D, Teel AR (2002) Open and closed loop dissipation inequalities under sampling and controller emulation. Eur J Control 18:109–125

Monaco S, Normand-Cyrot D (2007) Advanced tools for nonlinear sampled-data systems' analysis and control. Eur J Control 13:221–241

Nešić D, Teel AR (2004) A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. IEEE Trans Autom Control 49:1103–1034

Nešić D, Teel AR, Kokotović PV (1999) Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. Syst Control Lett 38:259–270

Nešić D, Teel AR, Carnevale D (2009) Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. IEEE Trans Autom Control 54: 619–624

Teel AR, Nešić D, Kokotović PV (1998) A note on input-to-state stability of sampled-data nonlinear systems. In: Proceedings of the conference on decision and control'98, Tampa, pp 2473–2478

Yuz JI, Goodwin GC (2005) On sampled-data models for nonlinear systems. IEEE Trans Autom Control 50:477–1488

N

Nonlinear System Identification Using Particle Filters

Thomas B. Schön
Department of Information Technology, Uppsala University, Uppsala, Sweden

Abstract

Particle filters are computational methods opening up for systematic inference in nonlinear/non-Gaussian state-space models. The particle filters constitute the most popular sequential Monte Carlo (SMC) methods. This is a relatively recent development, and the aim here is to provide a brief exposition of these SMC methods and how they are key enabling algorithms in solving nonlinear system identification problems. The particle filters are important for both frequentist (maximum likelihood) and Bayesian nonlinear system identification.

Keywords

Bayesian; Backward simulation; Maximum likelihood; Markov chain Monte Carlo (MCMC); Particle filter; Particle MCMC; Particle smoother; Sequential Monte Carlo

Introduction

The state-space model (SSM) offers a general tool for modeling and analyzing dynamical phenomena. The SSM consists of two stochastic processes: the states $\{\mathbf{x}_t\}_{t \geq 1}$ and the measurements $\{\mathbf{y}_t\}_{t \geq 1}$, which are related according to

$$\mathbf{x}_{t+1} \mid (\mathbf{x}_t = x_t) \sim f_\theta(x_{t+1} \mid x_t, u_t), \quad (1a)$$

$$\mathbf{y}_t \mid (\mathbf{x}_t = x_t) \sim h_\theta(y_t \mid x_t, u_t), \quad (1b)$$

and the initial state $\mathbf{x}_1 \sim \mu_\theta(x_1)$. We use bold face for random variables and $\sim$ means "distributed according to." The notation $\mathbf{x}_{t+1} \mid (\mathbf{x}_t = x_t)$ stands for the conditional probability of $\mathbf{x}_{t+1}$ given $\mathbf{x}_t = x_t$. The state process $\{\mathbf{x}_t\}_{t \geq 1}$ is a

Markov process, implying that we only need to condition on the most recent state $\mathbf{x_t}$, since that contains all information about the past. Furthermore, θ denotes the parameters, $f_\theta(\cdot)$ and $h_\theta(\cdot)$ that are probability density functions, encoding the dynamic and the measurement models, respectively. In the interest of a compact notation, we will suppress the input u_t throughout the text.

The SSM introduced in (1) is general in that it allows for nonlinear and non-Gaussian relationships. Furthermore, it includes both black-box and gray-box models on state-space form. Nonlinear black-box and gray-box models are covered by ▶ Nonlinear System Identification: An Overview of Common Approaches. The offline nonlinear system identification problem can (slightly simplified) be expressed as recovering information about the parameters θ based on the information in the T measured inputs $u_{1:T} \triangleq \{u_1, \ldots, u_T\}$ and outputs $y_{1:T}$. For a thorough exposition of the system identification problem, we refer to ▶ System Identification: An Overview. Nonlinear system identification has a long history, and a common assumption of the past has been that of linearity and Gaussianity. This assumption is very restrictive, and we have now witnessed well over half a century of research devoted to finding useful approximate algorithms allowing this assumption to be weakened. This development has significantly intensified during the past two decades of research on sequential Monte Carlo (SMC) methods (including particle filters and particle smoothers). However, the use of SMC for nonlinear system identification is more recent than that. The aim here is to introduce the key ideas enabling the use of SMC methods in solving nonlinear system identification problems, and as we will see, it is not a matter of straightforward application. The development of SMC-based identification follows two clear trends that are indeed more general: (1) The problems we are working with are analytically intractable, and hence, the mindset has to shift from searching for closed-form solutions to the use of *computational methods*, and (2) the new algorithms have basic building blocks that are themselves algorithms. Both these trends call for new developments.

Before the SMC methods are introduced in section "Sequential Monte Carlo", their need is clearly explained by formulating both the Bayesian and the maximum likelihood identification problems in sections "Bayesian Problem Formulation" and "Maximum Likelihood Problem Formulation", respectively. Solutions to these problems are then provided in sections "Bayesian Solutions" and "Maximum Likelihood Solutions", respectively. Finally, we give some intuition for online (recursive) solutions in section "Online Solutions", and in section "Summary and Future Directions", we conclude with a summary and directions for future research.

Bayesian Problem Formulation

In formulating the Bayesian problem, the parameters θ are modeled as unknown stochastic variables, i.e., the model (1) needs to be augmented with a prior density for the parameters $\boldsymbol{\theta} \sim p(\theta)$. The aim in Bayesian system identification is to compute the posterior density of $\boldsymbol{\theta}$ given the measurements $p(\theta \mid y_{1:T})$. More generally, we typically compute the joint posterior of the parameters $\boldsymbol{\theta}$ and the states $\mathbf{x}_{1:T}$,

$$p(\theta, x_{1:T} \mid y_{1:T}) = p(x_{1:T} \mid \theta, y_{1:T}) p(\theta \mid y_{1:T}). \tag{2}$$

By explicitly including the state variables $\mathbf{x}_{1:T}$ in the problem formulation according to (2), they take on the role of auxiliary variables. The reason for including the state variables $\mathbf{x}_{1:T}$ as auxiliary variables is that the alternative of excluding them would require us to analytically marginalize the states $\mathbf{x}_{1:T}$. This is not possible for the model (1) under study. However, once we have an approximation of $p(\theta, x_{1:T} \mid y_{1:T})$ available, the density $p(\theta \mid y_{1:T})$ is easily obtained by straightforward marginalization.

Maximum Likelihood Problem Formulation

In formulating the maximum likelihood (ML) problem, the parameters θ are modeled as unknown deterministic variables. The ML formulation offers a systematic way of computing point

estimates of the unknown parameters θ in a model, by making use of the information available in the obtained measurements $y_{1:T}$. The ML estimate is obtained by finding the θ that maximizes the so-called log-likelihood function, which is defined as

$$\ell_T(\theta) \triangleq \log p_\theta(y_{1:T}) = \sum_{t=1}^{T} \log p_\theta(y_t \mid y_{1:t-1}). \tag{3}$$

Note that we use θ as a subindex to denote that the corresponding probability density function is parameterized by θ, analogously to what was done in (1). The one step ahead predictor $p_\theta(y_t \mid y_{1:t-1})$ is computed by marginalizing $p(y_t, x_t \mid y_{1:t-1}) = h_\theta(y_t \mid x_t) p_\theta(x_t \mid y_{1:t-1})$ w.r.t. x_t, i.e., integrating out x_t from $p(y_t, x_t \mid y_{1:t-1})$. To summarize, the ML estimate $\hat{\theta}^{\text{ML}}$ is obtained by solving the following optimization problem:

$$\hat{\theta}^{\text{ML}} \triangleq \arg\max_{\theta} \ \sum_{t=1}^{T} \log \int h_\theta(y_t \mid x_t)$$
$$p_\theta(x_t \mid y_{1:t-1}) \mathrm{d}x_t. \tag{4}$$

This problem formulation clearly reveals the important fact that the nonlinear state inference problem (here computing $p_\theta(x_t \mid y_{1:t-1})$) is inherent in any maximum likelihood formulation for identification of SSMs. For linear Gaussian models, the Kalman filter offers closed-form solutions for the state inference problem, but for nonlinear models, there are no closed-form solutions available.

Sequential Monte Carlo

Solving the nonlinear system identification problem implicitly requires us to solve various nonlinear state inference problems. We will, for example, need to approximate the smoothing density $p(x_{1:T} \mid y_{1:T})$ and the filtering density $p(x_t \mid y_{1:t})$. The SMC samplers offer approximate solutions to these and other nonlinear state inference problems, where

the accuracy is only limited by the available computational resources. This section only deals with the state inference problem, allowing us to drop the θ in the notation for brevity.

Most SMC samplers hinge upon importance sampling, motivating section "Importance Sampling". In section "Particle Filter", we make use of importance sampling in computing an approximation of the filtering density $p(x_t \mid y_{1:t})$, and in section "Particle Smoother", a particle smoothing strategy is introduced to approximately compute $p(x_{1:T} \mid y_{1:T})$.

Importance Sampling

Let $\mathbf{z}$ be a random variable distributed according to some complicated density $\pi(z)$ and let $\varphi(\cdot)$ be some function of interest. Importance sampling offers a systematic way of evaluating integrals of the form

$$\mathrm{E}\left[\varphi(\mathbf{z})\right] = \int \varphi(z)\pi(z)\mathrm{d}z, \quad (5)$$

without requiring samples directly generated from $\pi(z)$. The density $\pi(z)$ is referred to as the *target* density, i.e., the density we are trying to sample from. The importance sampler relies on a *proposal* density $q(z)$, from which it is simple to generate samples, let $\mathbf{z}^i \sim q(z)$, $i = 1, \ldots, N$. Since each sample $\mathbf{z}^i$ is drawn from the proposal density rather than from the target density $\pi(z)$, we must somehow account for this discrepancy. The so-called importance weights $\tilde{\mathbf{w}}^i = \pi(\mathbf{z}^i)/q(\mathbf{z}^i)$ encode the difference. By normalizing the weights $\mathbf{w}^i = \tilde{\mathbf{w}}^i / \sum_{j=1}^{N} \tilde{\mathbf{w}}^j$, we obtain a set of weighted samples $\{\mathbf{z}^i, \mathbf{w}^i\}_{i=1}^{N}$ that can be used to approximately evaluate the integral (5) resulting in $\mathrm{E}\left[\varphi(\mathbf{z})\right] \approx \sum_{i=1}^{N} \mathbf{w}^i \varphi(\mathbf{z}^i)$. Schön and Lindsten (2014) provide an introduction to importance sampling within a dynamical systems setting, whereas Robert and Casella (2004) provide a general treatment.

Particle Filter

The solution to the nonlinear filtering problem is provided by the following two recursive equations:

$$p(x_t \mid y_{1:t}) = \frac{h(y_t \mid x_t)p(x_t \mid y_{1:t-1})}{p(y_t \mid y_{1:t-1})}, \quad (6a)$$

$$p(x_t \mid y_{1:t-1}) = \int f(x_t \mid x_{t-1})$$

$$p(x_{t-1} \mid y_{1:t-1})\mathrm{d}x_{t-1}. \quad (6b)$$

In the general case (1) there are no analytical solutions available for the above equations. The particle filter maintains an empirical approximation of the solution, which at time $t-1$ amounts to

$$\hat{p}^N(x_{t-1} \mid y_{1:t-1}) = \sum_{i=1}^{N} \mathbf{w}_{t-1}^i \delta_{\mathbf{x}_{t-1}^i}(x_{t-1}), \quad (7)$$

where $\delta_{\mathbf{x}_{t-1}^i}(x_{t-1})$ denotes the Dirac delta mass located at $\mathbf{x}_{t-1}^i$. Furthermore, $\mathbf{w}_{t-1}^i$ and $\mathbf{x}_{t-1}^i$ are referred to as the weights and the particles, respectively. We will now derive the particle filter by designing an importance sampler allowing us to approximately solve (6). The derivation is performed in an inductive fashion, starting by assuming that $p(x_{t-1} \mid y_{1:t-1})$ is approximated by (7). Inserting (7) into (6b) results in $\hat{p}^N(x_t \mid y_{1:t-1}) = \sum_{i=1}^{N} \mathbf{w}_{t-1}^i f(x_t \mid \mathbf{x}_{t-1}^i)$, which is used in (6a) to compute an *approximation* of the filtering density $p(x_t \mid y_{1:t})$ up to proportionality. Hence, this allows us to target $p(x_t \mid y_{1:t})$ using an importance sampler, where the form of $\hat{p}^N(x_t \mid y_{1:t-1})$ suggests that new samples can be proposed according to

$$\mathbf{x}_t^i \sim q(x_t \mid y_{1:t}) = \sum_{i=1}^{N} \mathbf{w}_{t-1}^i f(x_t \mid \mathbf{x}_{t-1}^i). \quad (8)$$

It is worth noting that we can obtain a more general algorithm by replacing $f(x_t \mid \mathbf{x}_{t-1}^i)$ in the above mixture with a density $q(x_t \mid \mathbf{x}_{t-1}^i, y_t)$. However, in the interest of a simple, but still highly useful algorithm, we keep (8). The proposal density (8) is a weighted mixture consisting of N components, which means that we can generate a sample $\tilde{\mathbf{x}}_t^i$ from it via a two-step procedure: first we select which component to sample from, and secondly we generate a sample from that component. More precisely, the first

Algorithm 1 Bootstrap particle filter (for $i = 1, \ldots, N$)
1. **Initialization ($t = 1$):**
   (a) Sample $\mathbf{x}_1^i \sim \mu(x_1)$.
   (b) Compute the importance weights $\widetilde{\mathbf{w}}_1^i = h(y_1 \mid \mathbf{x}_1^i)$ and normalize $\mathbf{w}_1^i = \widetilde{\mathbf{w}}_1^i / \sum_{j=1}^N \widetilde{\mathbf{w}}_1^j$.
2. **For $t = 2$ to T do:**
   (a) Resample $\{\mathbf{x}_{t-1}^i, \mathbf{w}_{t-1}^i\}$ resulting in equally weighted particles $\{\widetilde{\mathbf{x}}_{t-1}^i, 1/N\}$.
   (b) Sample $\mathbf{x}_t^i \sim f(x_t \mid \widetilde{\mathbf{x}}_{t-1}^i)$.
   (c) Compute the importance weights $\widetilde{\mathbf{w}}_t^i = h(y_t \mid \mathbf{x}_t^i)$ and normalize $\mathbf{w}_t^i = \widetilde{\mathbf{w}}_t^i / \sum_{j=1}^N \widetilde{\mathbf{w}}_t^j$.

part amounts to selecting one of the N particles $\{\mathbf{x}_{t-1}^i\}_{i=1}^N$ according to

$$\mathbb{P}\left(\widetilde{\mathbf{x}}_{t-1} = \mathbf{x}_{t-1}^i \mid \{\mathbf{x}_{t-1}^j, \mathbf{w}_{t-1}^j\}_{j=1}^N\right) = \mathbf{w}_{t-1}^i,$$

where the selected particle is denoted as $\widetilde{\mathbf{x}}_{t-1}$. By repeating this N times, we obtain a set of equally weighted particles $\{\widetilde{\mathbf{x}}_{t-1}^i\}_{i=1}^N$, constituting an empirical approximation of $p(x_{t-1} \mid y_{1:t-1})$, analogously to (7). We can then draw $\mathbf{x}_t^i \sim f(x_t \mid \widetilde{\mathbf{x}}_{t-1}^i)$ to generate a realization from the proposal (8). This procedure that turns a weighted set of samples into an unweighted one is commonly referred to as *resampling*.

Finally, using the approximation $\hat{p}^N(x_t \mid y_{1:t-1})$ in (6a) and the proposal density according to (8) allows us to compute the weights as $\widetilde{\mathbf{w}}_t^i = h(y_t \mid \mathbf{x}_t^i)$. Once all the N weights are computed and normalized, we obtain a collection of weighted particles $\{\mathbf{x}_t^i, \mathbf{w}_t^i\}_{i=1}^N$ targeting the filtering density at time t. We have now (in a slightly nonstandard fashion) derived the so-called *bootstrap particle filter*, which was the first particle filter introduced by Gordon et al. (1993) two decades ago. Since the introduction of Algorithm 1, the surrounding theory and practice have undergone significant developments; see, e.g., Doucet and Johansen (2011) for an up-to-date survey. The weights $\{\mathbf{w}_{1:T}^i\}_{i=1}^N$ and the particles $\{\mathbf{x}_{1:T}^i\}_{i=1}^N$ are random variables, and in executing the algorithm, we generate one realization from these. This is a useful insight both when it comes to understanding, but also when it comes to the analysis of the

particle filters. There is by now a fairly good understanding of the convergence properties of the particle filter; see, e.g., Doucet and Johansen (2011) for basic results and further pointers into the literature.

Particle Smoother

A particle smoother is an SMC method targeting the joint smoothing density $p(x_{1:T} \mid y_{1:T})$ (or one of its marginals). There are several different strategies for deriving particle smoothers. Rather than mentioning them all, we introduce one powerful and increasingly popular strategy based on *backward simulation*, giving rise to the family of *forward filtering/backward simulation* (FFBSi) samplers.

In an FFBSi sampler the joint smoothing density $p(x_{1:T} \mid y_{1:T})$ is targeted by complementing a forward particle filter with a second recursion evolving in the time-reversed direction. The following factorization of the joint smoothing density

$$p(x_{1:T} \mid y_{1:T}) = \left(\prod_{t=1}^{T-1} p(x_t \mid x_{t+1}, y_{1:t})\right) p(x_T \mid y_{1:T}),$$

immediately suggests a highly useful time-reversed recursion. Start by generating a sample $\widetilde{x}_T \sim p(x_T \mid y_{1:T})$. We then continue generating samples backward in time by sampling from the so-called backward kernel $p(x_t \mid x_{t+1}, y_{1:t})$ according to $\widetilde{x}_t \sim p(x_t \mid \widetilde{x}_{t+1}, y_{1:t})$, for $t = T - 1, \ldots, 1$. The resulting sample $\widetilde{\mathbf{x}}_{1:T} \triangleq (\widetilde{x}_1, \ldots, \widetilde{x}_T)$ is then by construction a sample from the joint smoothing density. Hence, in performing M backward simulations, we obtain the following approximation of the joint smoothing density:

$$\hat{p}^M(x_{1:T} \mid y_{1:T}) = \sum_{i=1}^M \frac{1}{M} \delta_{\widetilde{\mathbf{x}}_{1:T}^i}(x_{1:T}). \qquad (9)$$

For details on how to design algorithms implementing the backward simulation strategy,

derivations, properties, and references, we refer to the recent survey on backward simulation methods by Lindsten and Schön (2013).

Bayesian Solutions

Strategies

The posterior density (2) is analytically intractable, but we can make use of Markov chain Monte Carlo (MCMC) samplers to address the inference problem. An MCMC sampler allows us to approximately generate samples from an arbitrary target density $\pi(z)$. This is done by simulating a Markov chain (i.e., a Markov process) $\{z[r]\}_{r \geq 1}$, which is constructed in such a way that the stationary distribution of the chain is given by $\pi(z)$. The sample paths $\{z[r]\}_{r=1}^{R}$ of the chain can then be used to draw inference about the target distribution. Two *constructive* ways of finding a suitable Markov chain to simulate are provided by the Metropolis Hastings (MH) and the Gibbs samplers, where the latter can be interpreted as a special case of the former. See, e.g., Robert and Casella (2004) for details on MCMC. A Gibbs sampler targeting $p(\theta, x_{1:T} \mid y_{1:T})$ is given by

(i) Draw $\theta' \sim p(\theta \mid x_{1:T}, y_{1:T})$.

(ii) Draw $x'_{1:T} \sim p(x_{1:T} \mid \theta', y_{1:T})$.

The second step is hard, since it requires us to generate a sample from the joint smoothing density. Simply replacing step (ii) with a backward simulator does not result in a valid method (Andrieu et al. 2010).

One interesting solution is provided by the family of particle MCMC (PMCMC) sampler, first introduced by Andrieu et al. (2010). PMCMC provides a systematic way of combining SMC and MCMC, where SMC is used to construct the proposal density for the MCMC sampler. The so-called *particle Gibbs* (PG) sampler resolves the problems briefly mentioned above by a nontrivial modification of the SMC algorithm. Introducing the PG sampler lies outside the scope of this work; we refer the reader to the ground-breaking work by Andrieu et al. (2010). During the past 3 years, the PG samplers have developed quite a lot, and improved versions are surveyed and explained by Lindsten and Schön (2013).

A Nontrivial Example

To place PMCMC in the context of nonlinear system identification, we will now solve a nontrivial identification problem. The PG sampler is used to compute the posterior density for a general Wiener model (linear Gaussian system followed by a static nonlinearity) (Giri and Bai 2010):

$$\mathbf{x}_{t+1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{pmatrix} + \mathbf{v}_t, \qquad \mathbf{v}_t \sim \mathcal{N}(0, \mathbf{Q}), \tag{10a}$$

$$\mathbf{z}_t = \mathbf{C}\mathbf{x}_t, \tag{10b}$$

$$\mathbf{y}_t = \mathbf{g}(\mathbf{z}_t) + \mathbf{e}_t, \qquad \mathbf{e}_t \sim \mathcal{N}(0, \mathbf{r}). \tag{10c}$$

Based on observed inputs $u_{1:T}$ and outputs $y_{1:T}$, we wish to identify the model (10). We place a matrix normal inverse Wishart (MNIW) prior on $\{(\mathbf{A}, \mathbf{B}), \mathbf{Q}\}$, an inverse gamma prior on $\mathbf{r}$, and a Gaussian process (Rasmussen and Williams 2006) prior on the function $\mathbf{g}$, resulting in a semiparametric model. We can without loss of generality fix the matrix C according to $C = (1, 0, \dots, 0)$. For a complete model specification, we refer to Lindsten et al. (2013).

The posterior distribution $p(\theta, x_{1:T} \mid y_{1:T})$ is computed using a newly developed PG sampler referred to as particle Gibbs with ancestor sampling (PGAS); see Lindsten and Schön (2013). In the present experiment we make use of $T = 1,000$ observations. The dimension of the state–space is 6, the linear dynamics contains complex poles resulting in oscillations as seen in Fig. 1, and the nonlinearity is non-monotonic; see Fig. 2. A subspace method is used to find an initial guess for the linear system, and the static nonlinearity is initialized using a linear function (i.e., a straight line).

Nonlinear System Identification Using Particle Filters, Fig. 1
Bode diagram of the sixth-order linear system. The *black curve* is the true system. The *red curve* is the estimated posterior mean of the Bode diagram, and the *shaded area* is the 99 % Bayesian credibility interval
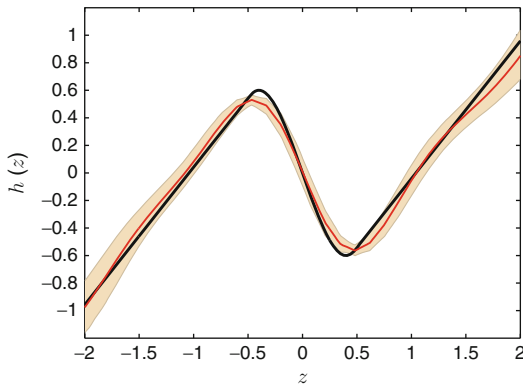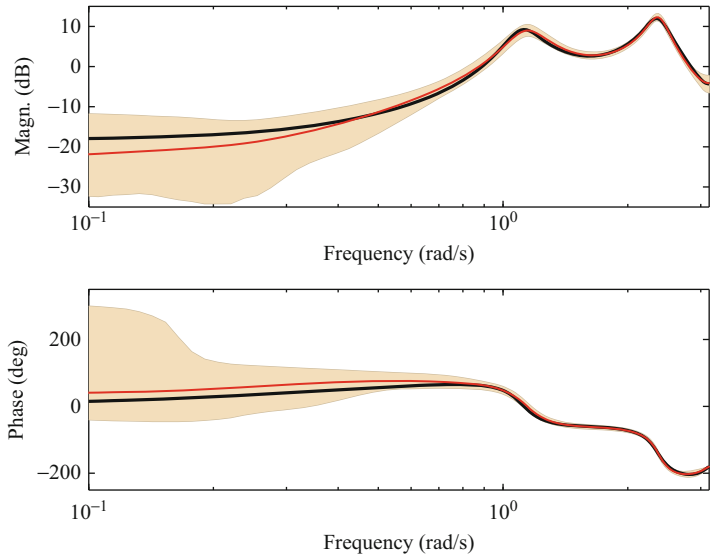
Nonlinear System Identification Using Particle Filters, Fig. 2 The *black curve* is the true static nonlinearity (non-monotonic). The *red curve* is the estimated posterior mean of the static nonlinearity, and the *shaded area* is the 99 % Bayesian credibility interval

It is worth pausing for a moment to reflect upon the posterior distribution $p(\theta, x_{1:T} \mid y_{1:T})$ that we are computing. The unknown "parameters" θ live in the space $\Theta = \mathbb{R}^{64} \times \mathcal{F}$, where $\mathcal{F}$ is an appropriate function space. The states $x_{1:T}$ live in the space $\mathbb{R}^{6 \times 1,000}$. Hence, $p(\theta, x_{1:T} \mid y_{1:T})$ is actually a rather complicated object for this example.

Using the PGAS sampler (with $N = 15$ particles), we construct a Markov chain $\{\theta[r], x_{1:T}[r]\}_{r=1}^{R}$ with $p(\theta, x_{1:T} \mid y_{1:T})$ as its stationary distribution. We run this Markov chain for $R = 25,000$ iterations, where the first 10,000 are discarded. The result is visualized in Figs. 1 and 2, where we plot the Bode diagram for the linear system and the static nonlinearity, respectively. In both figures we also provide the 99 % Bayesian credibility interval. MATLAB code for Bayesian identification of Wiener models is available from

The resonance peaks are accurately modeled, but the result is less accurate at low frequencies (likely due to a lack of excitation). The fact that the posterior mean is inaccurate at low frequencies is encoded in our estimate of the posterior distribution as shown by the credibility intervals.

In Figs. 1 and 2, we have visualized not only the posterior mean but also the uncertainty for the entire model. We could do this since the model is a linear dynamical system followed by a static nonlinearity. It would be most interesting if we

can come up with ways in which we could visualize the uncertainty inherent in general nonlinear dynamical systems.

Maximum Likelihood Solutions

Identifying the parameters θ in a general nonlinear SSM using maximum likelihood amounts to solving the optimization problem (3). This is a challenging problem for several reasons, for example, it requires the computation of the predictor density $p_\theta(y_t \mid y_{1:t-1})$. Furthermore, its gradient (possibly also its Hessian) is very useful in setting up an efficient optimization algorithm. There are no closed-form solutions available for these objects, forcing us to rely on approximations. The SMC methods briefly introduced in section "Sequential Monte Carlo" provide rather natural tools for this task, since they are capable of producing approximations where the accuracy is only limited by the available computational resources.

To establish a clear interface between the maximum likelihood problem (3) and the SMC methods, it has proven natural to make use of the expectation maximization (EM) algorithm (Dempster et al. 1977). The EM algorithm proceeds in an iterative fashion to compute ML estimates of unknown parameters θ in probabilistic models involving latent variables. The strategy underlying the EM algorithm is to exploit the structure inherent in the probabilistic model to separate the original problem into two closely linked problems. The first problem amounts to computing the so-called *intermediate quantity*

$$
\begin{aligned}
\mathcal{Q}(\theta, \theta') &\triangleq \int \log p_\theta(x_{1:T}, y_{1:T}) \\
&\quad p_{\theta'}(x_{1:T} \mid y_{1:T}) \mathrm{d}x_{1:T} \\
&= \mathrm{E}_{\theta'} \left[\log p_\theta(x_{1:T}, y_{1:T}) \mid y_{1:T} \right], \quad (11)
\end{aligned}
$$

where we have already made use of the fact that the latent variables in an SSM are given by the states. Furthermore, θ' denotes a particular value for the parameters θ. We can show that by choosing a new θ such that

$\mathcal{Q}(\theta, \theta') \geq \mathcal{Q}(\theta', \theta')$, the likelihood is either increased or left unchanged, i.e., $\ell_T(\theta) \geq \ell_T(\theta')$.

The EM algorithm now suggests itself in that we can generate a sequence of iterates $\{\theta^k\}_{k \geq 1}$ that guarantees that the log-likelihood is not decreased for increasing k by alternating the following two steps: (1) (Expectation) compute the intermediate quantity $\mathcal{Q}(\theta, \theta^k)$ and (2) (maximization) compute the subsequent iterate θ^{k+1} by maximizing $\mathcal{Q}(\theta, \theta^k)$ w.r.t. θ. This procedure is then repeated until convergence, guaranteeing convergence to a stationary point on the likelihood surface.

The FFBSi particle smoother offers an approximation of the joint smoothing density $p_{\theta'}(x_{1:T} \mid y_{1:T})$ according to (9), which inserted into (11) provides an approximative solution $\hat{\mathcal{Q}}^M(\theta, \theta')$ to the expectation step. In solving the maximization step, we typically want gradients of the intermediate quantity $\nabla_\theta \hat{\mathcal{Q}}^M(\theta, \theta')$. These can also be approximated using (9). The above development is summarized in Algorithm 2, providing a solution where the basic building blocks are themselves complex algorithms, an SMC algorithm for the E step and a nonlinear optimization algorithm for the M step. This means that we have the option of replacing the FFBSi particle smoother in step 2a with any other algorithm capable of producing estimates of the joint smoothing density. The family of PMCMC methods introduced in section "Bayesian Solutions" contains several highly interesting alternatives. A detailed account on Algorithm 2 is provided by Schön et al. (2011); see also Cappé et al. (2005).

Algorithm 2 EM for nonlinear system identification

1. **Initialization:** Set $k = 0$ and initialize θ^k.
2. **Expectation (E) step:**
   (a) Compute an approximation $\hat{p}_{\theta^k}^M(x_{1:T} \mid y_{1:T})$, for example, using an FFBSi sampler.
   (b) Calculate $\hat{\mathcal{Q}}^M(\theta, \theta^k)$.
3. **Maximization (M) step:** Compute

$$
\theta_{k+1} = \arg\max_\theta \hat{\mathcal{Q}}^M(\theta, \theta^k).
$$

4. Check termination condition. If satisfied, terminate; otherwise, update $k \to k + 1$ and return to step 2.

Finally, we mention Fisher's identity opening up yet another avenue for designing ML estimators using SMC approximations. Even if we are not interested in using EM when solving the nonlinear system identification problem, the intermediate quantity (11) is useful. The reason is provided via *Fisher's identity*,

$$\nabla_\theta \ell_T(\theta)\big|_{\theta=\theta'} = \nabla_\theta \mathcal{Q}(\theta, \theta')\big|_{\theta=\theta'}$$

$$= \int \nabla_\theta \log p_\theta(x_{1:T}, y_{1:T})\big|_{\theta=\theta'}$$
$$p_{\theta'}(x_{1:T} \mid y_{1:T}) \mathrm{d}x_{1:T},$$

which provides a means to compute approximations of the log-likelihood gradient. Hessian approximations are also available, but these are more involved. Hence, Fisher's identity opens up for direct use of any off-the-shelf gradient-based optimization method in solving (4).

Online Solutions

Online (also referred to as recursive or adaptive) identification refers to the problem where the parameter estimate is updated based on the parameter estimate at the previous time step and the new measurement. This is used when we are dealing with big data sets and in real-time situations. SMC offers interesting opportunities when it comes to deriving online solutions for nonlinear state- space models. The most direct idea is simply to make use of a gradient method

$$\theta_t = \theta_{t-1} + \gamma_t \nabla_\theta \log p_\theta(y_t \mid y_{1:t-1}),$$

where $\{\gamma_t\}$ is the sequence of step sizes. Fisher's identity (12) opens up for the use of SMC in approximating $\nabla_\theta \log p_\theta(y_t \mid y_{1:t-1})$ However, this leads to a rapidly increasing variance, something that can be dealt with by the so-called "marginal" Fisher identity; see Poyiadjis et al. (2011) for details.

An interesting alternative is provided by an online EM algorithm; see, e.g., Cappé (2011) for a solid introduction. The online EM approaches rely on the additive properties of the $\mathcal{Q}$-function. The area of online solutions via SMC is likely to grow in the future as there is a clear need motivated by the constantly growing data sets and there are also clear theoretical opportunities.

Summary and Future Directions

We have discussed how SMC samplers can be used to solve nonlinear system identification problems, by sketching both Bayesian and ML solutions. A common feature of the resulting algorithms is that they are (nontrivial) combinations of more basic algorithms. We have, for example, seen the combined use of a particle smoother and a nonlinear optimization solver in Algorithm 2 to compute ML estimates. As another example we have the class of PMCMC methods, where the basic building blocks are provided by SMC samplers and MCMC samplers. The use of SMC and MCMC methods for nonlinear system identification has only just started to take off, and it presents very interesting future prospects. Some directions for future research are as follows: (1) The family of PMCMC algorithms is rich and fast growing, with great potential for further developments. For example, its use in solving the state smoothing problem (i.e., computing $p(x_{1:T} \mid y_{1:T})$) is likely to provide better algorithms in the near future. (2) Related to this is the potential to design new particle smoothers capable of generating new particles also in the time-reversed direction. (3) There are open and highly relevant challenges when it comes to designing backward simulators for Bayesian nonparametric methods (Hjort et al. 2010). A key question here is how to represent the backward kernel $p(x_t \mid x_{t+1}, y_{1:t})$ in such nonparametric settings. (4) The use of Bayesian nonparametric models will open up interesting possibilities for hybrid system identification, since they allow us to systematically express and work with uncertainties over segmentations.

Cross-References

Recommended Reading

An overview of SMC methods for system identification is provided by Kantas et al. (2009), and a thorough introduction to SMC is provided by Doucet and Johansen (2011). The forthcoming monograph by Schön and Lindsten (2014) provides a textbook introduction to particle filters/smoothers (SMC), MCMC, PMCMC, and their use in solving problems in nonlinear system identification and nonlinear state inference. A self-contained introduction to particle smoothers and the backward simulation idea is provided by Lindsten and Schön (2013). The work by Cappé et al. (2005) also contains a lot of very relevant material in this respect.

Bibliography

Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain Monte Carlo methods. J R Stat Soc Ser B 72(2):1–33

Cappé O (2011) Online EM algorithm for hidden Markov models. J Comput Graph Stat 20(3): 728–749

Cappé O, Moulines E, Rydén T (2005) Inference in hidden Markov models. Springer, New York

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 39(1):1–38

Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) Nonlinear filtering handbook. Oxford University Press, Oxford, UK

Giri F, Bai E-W (eds) (2010) Block-oriented nonlinear system identification. Volume 404 of lecture notes in control and information sciences. Springer, Berlin/Heidelberg

Gordon NJ, Salmond DJ, Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc Radar Signal Process 140:107–113

Hjort N, Holmes C, Müller P, Walker S (eds) (2010) Bayesian nonparametrics. Cambridge University Press, Cambridge/New York

Kantas N, Doucet A, Singh S, Maciejowski J (2009) An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In: Proceedings of the 15th IFAC symposium on system identification, Saint-Malo, pp 774–785

Lindsten F, Schön TB (2013) Backward simulation methods for Monte Carlo statistical inference. Found Trends Mach Learn 6(1):1–143

Lindsten F, Schön TB, Jordan MI (2013) Bayesian semiparametric Wiener system identification. Automatica 49(7):2053–2063

Poyiadjis G, Doucet A, Singh SS (2011) Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. Biometrika 98(1):65–80

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT, Cambridge

Robert CP, Casella G (2004) Monte Carlo statistical methods, 2nd edn. Springer, New York

Schön TB, Lindsten F (2014) Learning of dynamical systems – particle filters and Markov chain methods. Forthcoming book, see user.it.uu.se/~thosc112/lds

Schön TB, Wills A, Ninness B (2011) System identification of nonlinear state-space models. Automatica 47(1):39–49

Nonlinear System Identification: An Overview of Common Approaches

Qinghua Zhang
Inria, Campus de Beaulieu, Rennes Cedex, France

Abstract

Nonlinear mathematical models are essential tools in various engineering and scientific domains, where more and more data are recorded by electronic devices. How to build nonlinear mathematical models essentially based on experimental data is the topic of this entry. Due to the large extent of the topic, this entry provides only a rough overview of some well-known results, from gray-box to black-box system identification.

Keywords

Black-box models; Block-oriented models; Gray-box models; Nonlinear system identification

Introduction

The wide success of linear system identification in various applications (Ljung 1999; ▶ System Identification: An Overview) does not necessarily mean that the underlying dynamic systems are

intrinsically linear. Quite often, linear system identification can be successfully applied to a nonlinear system if its working range is restricted to a neighborhood of some working point. Nevertheless, some advanced engineering systems may exhibit significant nonlinear behaviors under their normal working conditions, so do most biological or social systems. There is therefore an increasing demand on nonlinear dynamic system modeling theory. Nonlinear system identification is studied to partly answer this demand, when experimental data carry the essential information for modeling purpose.

Nonlinear system identification, compared to its linear counterpart, is a much more vast topic, as in principle a nonlinear model can be *any* description of a system which is not linear. For this reason, this entry provides only a rough overview of some well-known results.

An overview of the basic concepts of system identification can be found in ▶ System Identification: An Overview, notably the five basic elements to be taken into account in each application, among which the (nonlinear) model structures will be mainly focused on by this entry, as they represent the essential particularities of nonlinear system identification problems.

The various model structures used in nonlinear system identification are often classified by the level of available prior knowledge about the considered system: from *white-box* models to *black-box* models, via *gray-box* models. In principle, a white-box model is fully built from prior knowledge. Such a fully white-box approach is rarely feasible for complex systems because of insufficient prior knowledge or of intractable system complexity. Therefore, the system identification methods summarized in this entry concern gray-box and black-box models, for which experimental data play an essential role.

For ease of presentation, the main content of this entry will be restricted to the single-input single-output (SISO) case. The multiple-input multiple-output (MIMO) case will be discussed in the section "Multiple-Input Multiple-Output Systems" below.

Gray-Box Models

This section covers gray-box models, from the most to the least demanding ones in terms of prior knowledge.

Parametrized Physical Models

The dynamic behaviors of some engineering systems are governed by well-known physical laws, typically in the form of differential equations, possibly with unknown parameters. These parametrized physical equations can be used as gray-box models for system identification. In most situations, such a model can be written in the form of a vectorial first-order ordinary differential equation (ODE), known as *state equation,* and can be generally written as

$$\frac{dx(t)}{dt} = f(x(t), u(t); \theta) \qquad (1)$$

where t represents the time, $x(t)$ is the *state* vector, $u(t)$ the *input,* and $f(\cdot)$ a (nonlinear) function parametrized by the vector θ.

The observation on the system (typically with electronic sensors), referred to as the *output* and denoted by $y(t)$, is related to $x(t)$ and $u(t)$ through another known parametrized equation

$$y(t) = h(x(t), u(t); \theta) + v(t) \qquad (2)$$

where $v(t)$ represents the measurement error.

With digital electronic instruments, the input $u(t)$ and the output $y(t)$ are sampled at some discrete-time instants, say $t = \tau, 2\tau, 3\tau, \ldots, N\tau$ with some constant sampling period $\tau > 0$. For the sake of notation simplicity, let the sampling period $\tau = 1$ and assume ideal instantaneous samplers; then the sampled input-output data set is denoted by

$$Z^N = \{u(1), y(1), u(2), y(2), \ldots, u(N), y(N)\}. \qquad (3)$$

In some applications, data samples are made at irregular time instants. Some studies are particularly focused on system identification in this case (Garnier and Wang 2008).

The main remaining task of gray-box system identification is to estimate the parameter vector θ from the data set Z^N. The identification criterion is typically defined with the aid of an output predictor derived from the system model. A natural output predictor is simply based on the numerical solution of the state equation (1): for some given value of θ, initial state $x(0)$ and some assumed inter-sample behavior of the input $u(t)$ (e.g., with a zero order hold), the trajectory of $x(t)$, denoted by $\hat{x}(t|\theta)$, is computed with a numerical ODE solver, then the output prediction is computed as

$$\hat{y}(t|\theta) = h(\hat{x}(t|\theta), u(t); \theta). \qquad (4)$$

The parameter vector θ is typically estimated by minimizing the sum of squared prediction error $\varepsilon(t|\theta) = y(t) - \hat{y}(t|\theta)$. See ▶ System Identification: An Overview and Bohlin (2006) for more details.

The predictor based on the numerical solution of the state equation (1) (known as a *simulator*) may be in trouble if this equation with the given value of θ is unstable. Moreover, the state equation (1) may also be subject to some modeling error that should be taken into account in the output predictor. In such cases, the output predictor can be made with the aid of some nonlinear state observer (Gauthier and Kupka 2001) or some nonlinear filtering algorithm (Doucet and Johansen 2011).

Alternatively, sequential Monte Carlo (SMC) methods can also be applied to the identification of (small size) nonlinear state-space systems, typically assuming a discrete-time counterpart of the model described by Eqs. (1) and (2). See ▶ Nonlinear System Identification Using Particle Filters.

The gray-box approach is particularly useful in an engineering field when some software library of commonly used components is available. In this case, a system model can be built by connecting available component models. Nevertheless, the "connection" of the component models may introduce algebraic constraints through variables shared by connected components, leading to *differential algebraic equations* (DAE),

which are a wider class of dynamic system models than the abovementioned state-space models (▶ Modeling of Dynamic Systems from First Principles). For most dynamic systems, it is possible to avoid the DAE formulation by causality analysis, so that the connections between different system components are treated as information flow, instead of algebraic constraints. There exist also some theoretic studies on DAE system identifiability (Ljung and Glad 1994) and some recent developments on the identification of such systems (Gerdin et al. 2007).

Combined Physical and Black-Box Models

It may happen that, in a complex system, part of the components is well described by physical laws (possibly with available models from a software library), but some other components are not well studied. In this case, the latter components can be dealt with black-box models (or possibly empirical models). The entire model can be fitted to a collected data set Z^N, like in the case of the previous subsection.

Block-Oriented Models

Complex systems, notably those studied in engineering, are often made of a certain number of components; thus a system model can be built by connecting component models. In this sense, such component-based models could be said "block-oriented." In the system identification literature, the term *block-oriented model* is often used in a particular context (Giri and Bai 2010), where it is typically assumed that each component is either a *linear dynamic* subsystem or a *nonlinear static* one. Here, the term "static" means that the behavior of the component is memoryless and can be described by an algebraic equation. The study of system identification with such models is motivated by the fact that, when a controlled system is stabilized around a working point, its dynamic behavior can be well described by a linear model, but its actuators and sensors may exhibit significant nonlinear behaviors like saturation or dead zone. The choice of a particular block-oriented model structure depends on the prior knowledge about the underlying system,
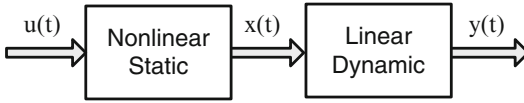
Nonlinear System Identification: An Overview of Common Approaches, Fig. 1 Hammerstein system
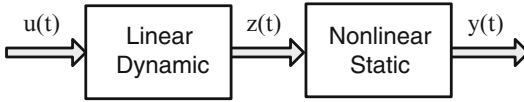
Nonlinear System Identification: An Overview of Common Approaches, Fig. 2 Wiener system

with specific identification methods available for different model structures.

The most frequently studied block-oriented models for system identification concern the Hammerstein system and the Wiener system, each composed of two blocks, as illustrated, respectively, in Figs. 1 and 2.

Hammerstein System Identification

A SISO Hammerstein system is typically formulated as

$$x(t) = f(u(t)) \tag{5a}$$

$$
\begin{aligned}
y(t) &+ a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) \\
&= b_1 x(t-1) + \cdots + b_{n_b} x(t-n_b) \\
&+ v(t).
\end{aligned} \tag{5b}
$$

If the nonlinearity $f(\cdot)$ is expressed in the form of

$$f(u) = \sum_{l=1}^{m} \gamma_l \kappa_l(u) \tag{6}$$

with some chosen basis functions $\kappa_l(\cdot)$, then the identification problem amounts to fitting the model parameters γ_l, a_i, b_j to a collected data set Z^N. A well-known method is based on over-parametrization (Bai 1998): replace in (5b) each $x(t-j)$ with the right-hand side of (6) and treat each parameter product $b_j \gamma_l$ as an individual parameter, then the newly parametrized model is equivalent to a linear ARX model (▶ System Identification: An Overview), which can be estimated by a well-established linear system identification method. As the $n_b + m$ parameters b_j and γ_l; are replaced by $n_b m$ parameters in the new parametrization, the term "over-parametrization" refers to the fact that typically $n_b + m < n_b m$. The estimated over-parametrized model can be reduced to the original parametrization, usually through the singular value decomposition (SVD) of the matrix filled with the estimated parameter products $b_j \gamma_l$. See Giri and Bai (2010) for other identification methods with variant formulations of Hammerstein system model.

When the linear subsystem is approximated by a finite impulse response (FIR) model, it is possible to first estimate the linear model before estimating a model for the nonlinear block (Greblicki and Pawlak 1989).

Wiener System Identification

A SISO Wiener system is typically formulated as

$$z(t) = \sum_{k=1}^{\infty} h_k u(t-k) \tag{7a}$$

$$y(t) = g(z(t)) + v(t) \tag{7b}$$

where the sequence $h_1, h_2, \ldots$ is the impulse response of the linear subsystem, $g(\cdot)$ is some nonlinear function, and $v(t)$ is a noise independent of the input $u(t)$.

Some methods for Wiener system identification assume a finite impulse response (FIR) of the linear subsystem. In this case, the linear subsystem model is characterized by the vector collecting the FIR coefficients $h^T = [h_1, h_2, \ldots, h_n]$. There are two typical kinds of efficient solutions, assuming either the Gaussian distribution of the input $u(t)$ (Greblicki 1992) or the monotonicity of the nonlinear function $g(\cdot)$ (Bai and Reyland Jr 2008). In both cases, it is possible to directly estimate the FIR coefficients h from the input-output data Z^N, without explicitly estimating the unknown nonlinear function $g(\cdot)$. The estimated h can be used to compute the internal variable $z(t)$. It then becomes relatively easy to estimate the nonlinear function $g(\cdot)$ from the computed $z(t)$ and the measured $y(t)$.

Other Block-Oriented Model Structures

Among block-oriented models composed of more blocks, the most well-known ones concern Hammerstein-Wiener system and Wiener-Hammerstein system. They are both composed of 3 blocks connected in series, the former has a linear dynamic block preceded and followed by two nonlinear static blocks, and the latter has a nonlinear static block in the middle of two linear dynamic blocks. In general, the prediction error method (PEM) (Ljung 1999) is applied to the identification of such systems, with heuristic methods for the initialization of model parameters. Some recent results on Hammerstein-Wiener system identification have been reported in Wills et al. (2013). There exist also some other variants, with parallel blocks or feedback loops. In most cases, each block is either *linear dynamic* or *nonlinear static,* but there is a notable exception: *hysteresis* blocks. Hysteresis is a phenomenon typically observed in some magnetic or mechanic systems. Its mathematical description is both dynamic and strongly nonlinear and cannot be decomposed into linear dynamic and nonlinear static blocks. Due to the importance of hysteresis components in some control systems, system identification involving such blocks is currently an active research topic (Giri et al. 2008).

LPV Models

Linear parameter-varying (LPV) models could be classified as black-box models, because typically they rely more on experimental data than on prior knowledge. However, engineers often have good insights into such models; they are thus presented in the gray-box section.

From Gain Scheduling to LPV Models

Gain scheduling is a method originally developed for the control of nonlinear systems. It consists in designing different controllers for different working points of a nonlinear system and in switching among the designed controllers according to the actual working point. It is typically assumed that the working point is determined by some observed variable (vector) referred to as the *scheduling variable* and denoted by ρ. Around

each considered working point, the nonlinear system is linearized so that the corresponding controller can be designed from the linear control theory. A by-product of this controller design procedure is a collection of linearized models indexed by the scheduling variable ρ. This collection, seen as a whole model of the globally nonlinear system, is known as an LPV model (Toth 2010). This approach has been particularly successful in the field of flight control.

An LPV model can be formulated either in input-output form or in state-space form. In the input-output form, a SISO model can be written as

$$y(t) + a_1(\rho) y(t-1) + \cdots + a_{n_a}(\rho) y(t - n_a)$$
$$= b_1(\rho) u(t-1) + \cdots + b_{n_b}(\rho) u(t - n_b)$$
$$+ v(t). \tag{8}$$

and in the state-space form as

$$x(t+1) = A(\rho) x(t) + B(\rho) u(t) + w(t) \tag{9a}$$

$$y(t) = c(\rho) x(t) + D(\rho) u(t) + v(t) \tag{9b}$$

As a global model of the whole nonlinear system, the ρ-dependent parameters (matrices) $a_i(\rho), b_j(\rho), A(\rho)$, etc., are functions defined for all $\rho \in \Omega$, where Ω is the relevant working range of the considered system (a compact subset of a real vector space). If originally the LPV model was built through a collection of linearized models around different working points, then the values of these functions are first defined for the corresponding discrete values of ρ. For other values of $\rho \in \Omega$, these functions can be defined by interpolation. Alternatively, by choosing some parametric forms of $a_i(\rho), b_j(\rho), A(\rho)$, etc., the whole LPV model can also be estimated by fitting it to a data set Z^N, through nonlinear optimization (Toth 2010).

Local Linear Models

In an LPV model, the model parameters can in principle depend on the scheduling variable ρ in

any chosen manner. A particularly useful case is when they are formulated as expansions over local basis functions. For example, in (8), the parameter $a_i(\rho)$ may be expressed as

$$a_i(\rho) = \sum_{l=1}^{m} a_{i,l}\kappa_l(\rho) \qquad (10)$$

where $k_l(\cdot)$ are some chosen bell-shaped (local) basis functions, typically the Gaussian function, centered at different positions $\rho = c_{l,} \in \Omega$, and $a_{i,l}$ are coefficients of the expansion. Similarly

$$b_j(\rho) = \sum_{l=1}^{m} b_{i,l}\kappa_l(\rho). \qquad (11)$$

Assume that the basis functions are normalized such that

$$\sum_{l=1}^{m} \kappa_l(\rho) = 1 \qquad (12)$$

for all $\rho \in \Omega$. Then the LPV model (8) can be viewed as an interpolation of m "local" models

$$y(t) + a_{1,l}y(t-1) + \cdots + a_{n_a,l}y(t-n_a)$$
$$= b_{1,l}u(t-1) + \cdots + b_{n_b,l}u(t-n_b) + v(t) \qquad (13)$$

indexed by $l = 1, 2, \ldots, m$. Each of these linear models is valid for ρ close to c_l, the center of the corresponding basis function $\kappa_l(\cdot)$; hence, they are called *local linear models*.

If the local basis functions $\kappa_l(\rho)$ are viewed as *membership functions* of fuzzy sets, then the local linear model is strongly related to the Takagi-Sugeno fuzzy model (Takagi and Sugeno 1985). An advantage of this point of view is the possibilities of incorporating prior knowledge in the form of linguistic rules and of interpreting some local linear models resulting from system identification.

There are two approaches to building local linear models. The first one is the local approach: for each chosen value of $c_{l,} \in \Omega$, a local model is estimated from data corresponding to the values of ρ within a neighborhood of c_l. This approach has the advantages of being computationally efficient, easily updatable, and well understood by engineers. The second one

is the global approach: all the model parameters are estimated simultaneously by solving a single optimization problem for the whole model. This approach can produce more accurate models in terms of prediction error, but it is numerically much more expensive and may lead to models difficult to be interpreted by engineers.

The practical success of local linear models strongly depends on the possibility of finding a scheduling variable ρ of small dimension relevantly determining the working point of the considered system. If there exists a nonlinear state-space model of the system, then in principle the working point is determined jointly by the state and the input of the system. As quite often physically meaningful state variables are not fully observed, they cannot be used in the definition of ρ. It is possible to define ρ as delayed output and input variables, e.g.,

$$\rho^T = [y(t), \ldots, y(t-n_a), u(t-1), \ldots, u(t-n_b)]$$

but it typically leads to a vector of quite large dimension. It is thus important to use practical insights about a given system to find a relevant vector ρ of reduced dimension.

For a single-dimensional ρ, the choice of the local basis function centers c_l can be made following some practical insight or equally spaced within Ω. For a large-dimensional ρ, this task is more difficult. The equally spaced approach would lead to too many local models, as their number would exponentially increase with the dimension of ρ. In this case, an empirical approach, called *local linear model tree* (LOLIMOT) (Nelles 2001), can be applied. It iteratively partitions Ω in order to place the local basis functions where the system is more likely nonlinear or where the available data are more concentrated.

Black-Box Models

Ideally speaking, a black-box model should be solely built from experimental data, without any prior knowledge. In practice, some prior knowledge is always necessary, though experimental

data play a much more important role. For instance, the choice of the input and output variables, implying some causality relationship, is an important prior knowledge.

With the fast development of electronic devices, more and more sensor signals are available in various fields, notably for engineering, environmental, and biomedical systems. Meanwhile, the processing power of modern computers increases every year. Black-box modeling has thus more and more potential applications. Nevertheless, the importance of prior knowledge in a modeling procedure should not be forgotten. In general prior knowledge leads to more reliable models in terms of validity range, as the validity of physical equations is often well understood. In contrast, for a black-box model essentially based on experimental data, it may be hard to ensure its validity for interpolation and even harder for extrapolation.

Input-Output Black-Box Models

As the primary role of a mathematical model is to predict the output of the system for given input values, it is natural to design black-box models directly in the form of a predictor. As the output $y(t)$ of a dynamic system depends on the past inputs, a predicted output $\hat{y}(t)$ may be formulated in the form of

$$\hat{y}(t) = f(u(t-1), u(t-2), \ldots, u(t-n_b)) \tag{14}$$

where $f(\cdot)$ is some nonlinear function (to be estimated from experimental data) and n_b is a chosen integer. In principle, n_b can be infinitely large (as $y(t)$ depends on *all* the past inputs in general), but in practice, a model of finite complexity has to be chosen. If the considered system is stable in the sense that sufficiently old past inputs are (gradually) forgotten, then it is reasonable to truncate the dependence on the past inputs.

The model structure (14) is similar to the linear finite impulse response (FIR) model (Ljung 1999). It is known that, for linear system identification, the use of ARX models, predicting $y(t)$ from both past inputs and past outputs, is often

more efficient than FIR models, in the sense of requiring fewer model parameters. By analogy, the nonlinear ARX model takes the form

$$\hat{y}(t) = f(y(t-1), \ldots, y(t-n_a),$$
$$u(t-1), \ldots, u(t-n_b)). \tag{15}$$

This is likely the most frequently used black-box model structure for nonlinear dynamic system identification (Sjöberg et al. 1995; Juditsky et al. 1995).

Nonlinear Function Estimators

For a nonlinear ARX model in the form of (15), the nonlinear function $f(\cdot)$ has to be estimated from an available input-output data set Z^N. Typically, an estimator of $f(\cdot)$ with some chosen parametric structure is used. Let

$$\phi^T(t) = [y(t-1), \ldots, y(t-n_a),$$
$$u(t-1), \ldots, u(t-n_b)], \tag{16}$$

then system identification in this case amounts to solving a nonlinear regression problem

$$y(t) = g(\phi(t); \theta) + v(t) \tag{17}$$

where $g(\cdot)$ is a chosen nonlinear function parametrized by θ, capable of approximating a large class of nonlinear functions by appropriately adjusting θ, and $v(t)$ is the modeling error to be minimized in some sense.

The most well-known nonlinear function estimators implementing $g(\cdot)$ in practice are polynomials, splines, multiple-layer neural networks, radial basis networks, wavelets, and fuzzy-neural estimators. Most of these estimators can be written in the form

$$g(\phi(t); \theta) = \sum_{l=1}^{m} \gamma_l \kappa(\alpha_l(\phi - \beta_l)) \tag{18}$$

or in some close variant of this form, where $\kappa(\cdot)$ is some "mother" basis function dilated and translated by α_l and β_l before being weighted by γ_l in the sum forming the estimator (Sjöberg et al. 1995). For example, $\kappa(\cdot)$ is

typically chosen as a (Gaussian) bell-shaped function in radial basis networks or a sigmoid (S-shaped) function in multiple-layer neural networks.

Another approach to nonlinear function estimation is called *nonparametric estimation.* Its main idea is to estimate $f(\varphi^*)$ for any given value of φ^* by the (weighted) average of the values of $y(t)$ in the available data set corresponding to values of $\varphi(t)$ close to φ^*. This category includes *kernel* estimators (Nadaraya 1964) and *memory-based* estimators (Specht 1991).

The nonlinear function estimation problem as formulated in (17) can also be addressed with the Gaussian process model. Assume that g in (17) is a Gaussian process whose covariance matrix for any regressor pair $\varphi(t)$ and $\varphi(\tau)$ is a known function of the regressor pair, then the posterior distribution of g given observations on $y(t)$ can be computed by applying the Bayes' theorem under certain assumptions (Rasmussen and Williams 2006). This method is strongly related to the least squares support vector machines (Suykens et al. 2002) and to some extent is similar to kernel estimators.

The difficulty for estimating a nonlinear function $f(\varphi)$ strongly depends on the dimension of φ. In the single-dimensional case, most existing methods can produce satisfactory results. When the dimension of φ, say n, increases, in order to keep the data "density" unchanged, the number of data points must increase exponentially with n. This fact implies that, in the high-dimensional case (say $n > 10$), for most practically available data sets, the data points are sparse in the space of φ. It is thus practically impossible to estimate $f(\varphi)$ with a good accuracy everywhere in the space of φ. In order to remedy this problem, prior knowledge can be used to form a more elaborated vector φ of reduced dimension, instead of the simple form of past input and output variables. The resulting model will be more of gray-box nature. If this approach is not possible, one has to expect that the estimation algorithm automatically discovers some low-dimension nature of the nonlinear relationship being estimated. The success would depend on the suitability of the

chosen particular nonlinear function estimator for the considered system.

State-Space Black-Box Models

For a gray-box model in the form of (1) and (2), it is assumed that the parametric forms of the nonlinear functions $f(\cdot)$ and $h(\cdot)$ are known from prior knowledge. If no such knowledge is available, it is possible to estimate these nonlinear functions with some function estimator, like those introduced in the previous subsection. Such an approach leads to state-space black-box models. In practice, it is easier to use the discrete-time counterpart of the state equation (1). Because typically the state vector $x(t)$ is not directly observed, the estimation of $f(\cdot)$ and $h(\cdot)$ cannot be formulated as nonlinear regression problems, in contrast to the case of input-output black-box models. Another difficulty is related to the nonuniqueness of the state-space representation of a given system: any (linear or nonlinear) state transformation would lead to a different state-space representation of the same system. In some existing methods, a linear state-space model is first estimated; then nonlinear function estimators are used to compensate the residuals of $f(\cdot)$ and $h(\cdot)$ after their linear approximations (Paduart et al. 2010).

Multiple-Input Multiple-Output Systems

For multiple-input multiple-output (MIMO) systems, state-space models like (1)–(2) remain in the same form, by considering vector values of the notations $u(t)$ and $y(t)$ at each time instant, up to some similar adaptation of the other involved notations. For input-output models like (15), the involved notations can also be vector valued, but the fact that different inputs and/or outputs can have different delays makes the notations more complicated. For block-oriented models, though a MIMO linear block is usually described by a general linear model in state-space form or in input-output form, there is no consensus for the structural choice of MIMO nonlinear blocks.

Some Practical Issues

The general practical aspects discussed in ▶ System Identification: An Overview are of course also valid for nonlinear system identification, but some particularities in the nonlinear case should be highlighted.

It is important to apply appropriate input signals so that the collected data convey sufficient information for system identification. The design of input signals for this purpose is known as *experiment design*. In the framework of linear system identification, experiment design is usually formulated through the optimization of the covariance matrix of model parameter estimates (▶ Experiment Design and Identification for Control), which often leads to non-convex optimization problems. Experiment design in the nonlinear case has not been systematically studied. If possible, the chosen input signal should be similar to what will be actually applied to the considered system and cover various working conditions. Another simple rule is that the input should excite a nonlinear system at different amplitudes, whereas binary input signals are often used for linear systems.

Model validation is a particularly delicate task for nonlinear black-box models. As already mentioned when such models are introduced, the available data points are usually sparse when a nonlinear function is estimated in a high-dimensional space; it is thus practically impossible to uniformly ensure the estimation accuracy of the nonlinear function. It is important to extensively perform *cross-validation,* by testing the validity of the model on large data sets that have not been used for model estimation.

Regularization is also an important issue for nonlinear black-box models. Because of lack of prior knowledge, each nonlinear black-box model has a flexible structure in order to cover a large class of nonlinear systems, typically with many model parameters, implying large variances of parameter estimates (▶ System Identification: An Overview). Appropriately applying a regularized criterion for model parameter estimation can reduce the variances. For gray-box models, prior knowledge can be used for regularization through a Bayesian approach, but this approach is not applicable to black-box models.

Summary and Future Directions

Compared to linear system identification, the nonlinear case is a much more vast topic, of which this entry provides only a rough overview. The main lines that should be retained are that both prior knowledge and experimental data are required for system identification and that the more prior knowledge is incorporated in a model, the better the extent of its validity is understood. The lack of prior knowledge should be compensated by the processing of large amounts of data. The data that can be processed within an acceptable time depend on the power of computers that progresses every year. Meanwhile, the research and development of efficient algorithms for large data processing with multiple or massively parallel processors are an exciting topic in system identification.

Cross-References

- ▶ Experiment Design and Identification for Control
- ▶ Modeling of Dynamic Systems from First Principles
- ▶ Nonlinear System Identification Using Particle Filters
- ▶ System Identification: An Overview

Recommended Reading

Nonlinear system identification is covered by a vast literature. After the readings about general topics on system identification (see ▶ System Identification: An Overview and references therein), the reader may further read (Nelles 2001) for black-box system identification, (Bohlin 2006) for gray-box system identification, (Giri and Bai 2010) for block-oriented system identification, and (Toth 2010) for LPV system identification.

Bibliography

Bai EW (1998) An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems. Automatica 34(3):333–338

Bai E-W, Reyland Jr J (2008) Towards identification of Wiener systems with the least amount of a priori information on the nonlinearity. Automatica 44(4):910–919

Bohlin T (2006) Practical grey-box process identification – theory and applications. Springer, London

Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) Nonlinear filtering handbook. Oxford University Press, Oxford

Garnier H, Wang L (eds) (2008) Identification of continuous-time models from sampled data. Springer, London

Gauthier J-P, Kupka I (2001) Deterministic observation theory and applications. Cambridge University Press, Cambridge/New York

Gerdin M, Schön T, Glad T, Gustafsson F, Ljung L (2007) On parameter and state estimation for linear differential-algebraic equations. Automatica 43:416–425

Giri F, Bai E-W (eds) (2010) Block-oriented nonlinear system identification Springer, Berlin/Heidelberg

Giri F, Rochdi Y, Chaoui FZ, Brouri A (2008) Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities. Automatica 44(3):767–775

Greblicki W (1992) Nonparametric identification of Wiener systems. IEEE Trans Inf Theory 38(5):1487–1493

Greblicki W, Pawlak M (1989) Nonparametric identification of Hammerstein systems. IEEE Trans Inf Theory 35(2):409–418

Juditsky A, Hjalmarsson H, Benveniste A, Delyon B, Ljung L, Sjöberg J, Zhang Q (1995) Nonlinear black-box models in system identification: mathematical foundations. Automatica 31(11):1725–1750

Ljung L (1999) System identification – theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River

Ljung L, Glad T (1994) On global identifiability for arbitrary model parametrizations. Automatica 30(2):265–276

Nadaraya EA (1964) On estimating regression. Theory Probab Appl 9:141–142

Nelles O (2001) Nonlinear system identification. Springer, Berlin/New York

Paduart J, Lauwers L, Swevers J, Smolders K, Schoukens J, Pintelon R (2010) Identification of nonlinear systems using polynomial nonlinear state space models. Automatica 46(4):647–656

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT, Cambridge

Sjöberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec P-Y, Hjalmarsson H, Juditsky A (1995) Non-linear black-box modeling in system identification: a unified overview. Automatica 31(11):1691–1724

Specht DF (1991) A general regression neural network. IEEE Trans Neural Netw 2(5):568–576

Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific, Singapore

Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 15(1):116–132

Toth R (2010) Modeling and identification of linear parameter-varying Systems. Springer, Berlin

Wills A, Schön T, Ljung L, Ninness B (2013) Identification of Hammerstein-Wiener models. Automatica 49(1):70–81

Nonlinear Zero Dynamics

Alberto Isidori
Department of Computer and System Sciences
"A. Ruberti", University of Rome
"La Sapienza", Rome, Italy

Abstract

The notion of zero dynamics plays a role in nonlinear systems that is analogous to the role played, in a linear system, by the notion of zeros of the transfer function. In this article, we review the basic concepts underlying the definition of zero dynamics and discuss its relevance in the context of nonlinear feedback design.

Keywords

High-gain feedback; Inverse systems; Minimum-phase nonlinear systems; Normal forms; Output regulation; Stabilization

Introduction

The concept of zero dynamics of a nonlinear system was introduced in the early 1980s as the nonlinear analogue of the concept of transmission zero of a linear system. This concept played a fundamental role in the development of systematic methods for asymptotic stabilization of

relevant classes of nonlinear systems. As a matter of fact, a nonlinear system in which the zero dynamics possess a globally asymptotically stable equilibrium can be robustly stabilized, globally or at least with guaranteed region of attraction, by means of output feedback. This is a nonlinear analogue of a well-know property of linear systems, namely, the property that an n-dimensional linear systems having $n - 1$ zeros with negative real part can be stabilized by means of proportional output feedback, if the feedback gain is sufficiently large. The concept of zero dynamics also plays a relevant role in variety of other problems of feedback design, such as input-output linearization with internal stability, non-interacting control with internal stability, output regulation, and feedback equivalence to passive systems.

The Zero Dynamics

One of the cornerstones of the geometric theory of control systems (for linear as well as for nonlinear systems) is the analysis of how the observability property can be influenced by feedback. This study, originally conceived in the context of the problem of disturbance decoupling, had far reaching consequences in a number of other domains. One of these consequences is the possibility of characterizing in "geometric terms" the notion of *zero* of the transfer function of a system. In a (single-input single-output and minimal) linear system, a complex number z is a zero of the transfer function if and only if the input $u(t) = \exp(zt)$ yields – for a suitable choice of the initial state – a forced response in which the output is identically zero. This "open-loop" and "time-domain" characterization has a "closed-loop and "geometric" counterpart: all such z's coincide with the eigenvalues of the unobservable part of the system, once the latter has been rendered maximally unobservable by means of feedback. One of the earlier successes of the geometric approach to the analysis and design of nonlinear systems was the possibility of extending these equivalent characterizations to the domain of nonlinear systems.

To see how this is possible, consider for simplicity the case of a system modeled by equations of the form

$$
\begin{aligned}
\dot{x} &= f(x) + g(x)u \\
y &= h(x)
\end{aligned}
$$

with state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}$, output $y \in \mathbb{R}$ and in which $f(x), g(x), h(x)$ are smooth functions. Systems of this forms are called *input-affine systems*. The analysis of such systems is rendered particularly simple if appropriate notations are used. Given any real-valued smooth function $\lambda(x)$ and any n-vector valued smooth function $X(x)$, let $L_X \lambda(x)$ denote the (directional) derivative of $\lambda(x)$ along $X(x)$, that is the real-valued smooth function

$$
L_X \lambda(x) = \sum_{i=1}^{n} \frac{\partial \lambda}{\partial x_i} X_i(x) ,
$$

and, recursively, set $L_X^d \lambda = L_X L_X^{d-1} \lambda(x)$ for any $d \geq 1$.

Suppose there exists an integer $r \geq 1$ with the following properties

$$
L_g h(x) = L_g L_f h(x) = \cdots = L_g L_f^{r-2} h(x)
$$

$$
= 0 \quad \forall x \in \mathbb{R}^n
$$

$$
L_g L_f^{r-1} h(x) \neq 0 \qquad \forall x \in \mathbb{R}^n .
$$

If this is the case, it is possible to show that the set

$$
Z^* = \{x \in \mathbb{R}^n : h(x) = L_h(x) = \cdots
$$

$$
= L_f^{r-1} h(x) = 0\}
$$

is a smooth sub-manifold of $\mathbb{R}^n$, of codimension r. It is also easy to show that the state-feedback law

$$
u^*(x) = -\frac{L_f^r h(x)}{L_g L_f^{r-1} h(x)}
$$

renders the vector

$$
f^*(x) = f(x) + g(x)u^*(x)
$$

tangent to Z^*, at each point x of Z^*. In other words, Z^* is an *invariant* manifold of the feedback-modified system

$$\dot{x} = f^*(x).$$

It is seen from this construction that the output $y(t) = h(x(t))$ of the system is identically zero if and only if $x(0) \in Z^*$ and $u(t) = u^*(x(t))$, where $x(t)$ is the solution of $\dot{x} = f^*(x)$ passing through $x(0)$ at time $t = 0$. As a consequence, the *restriction* of $\dot{x} = f^*(x)$ to its invariant manifold Z^* characterizes all *internal* dynamics that occur in the system once initial condition and input are chosen in such a way that the output is constrained to be identically zero. The dynamics in question are called the *zero-dynamics* of the system. Note that this construction demonstrates, as anticipated, the equivalence between an "open-loop" and a "closed-loop" characterization of all the (internal) dynamics of a given system that are compatible with the constraint that the output is identically zero. This construction can be extended to multi-input multi-output systems, with the aid of an appropriate recursive algorithm, known as the *zero dynamics algorithm* (Isidori 1995).

Normal Forms

The coordinate-free construction presented above becomes even more transparent if special coordinates are chosen. To this end, set

$$g^*(x) = \frac{1}{L_g L_f^{r-1} h(x)} g(x)$$

and define, recursively,

$$X_0(x) = g^*(x), \quad X_k(x) = [f^*(x), X_{k-1}(x)],$$

for $1 \le k \le r-1$, in which $[Y(x), X(x)]$ denotes the Lie bracket of $Y(x)$ and $X(x)$. It is possible to show that if the vector fields $X_0(x), \dots, X_{n-1}(x)$ are *complete*, there exists a smooth nonlinear, *globally defined*, change of variables by means of which the system can be transformed into a system of the form

$$\dot{z} = f_0(z, \xi)$$
$$\dot{\xi} = A_r \xi + B_r [q_0(z, \xi) + b(z, \xi)u]$$
$$y = C_r \xi$$

in which $z \in \mathbb{R}^{n-r}$, $\xi \in \mathbb{R}^n$, the matrices A_r, B_r, C_r have the form

$$A_r = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \quad B_r = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \\ 1 \end{pmatrix},$$

$$C_r = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

and $b(z, \xi) \ne 0$ for all (z, ξ). These equations are said to be in *normal form* (Isidori 1995).

It is easy to check that, in these coordinates, the manifold Z^* is the set of all pairs (z, ξ) having $\xi = 0$, the state feedback law $u^*(x)$ is the function

$$u^*(z, \xi) = -\frac{q_0(z, \xi)}{b(z, \xi)}$$

and the restriction of $\dot{x} = f^*(x)$ to the manifold Z^* is nothing else than

$$\dot{z} = f_0(z, 0).$$

The latter provide a simple characterization of the zero dynamics of the system, once that the latter has been brought to its normal form.

It is worth observing that, in the case of a linear system, functions $f_0(z, \xi)$ and $q_0(z, \xi)$ are linear functions, and $b(z, \xi)$ is a constant. Consequently, the normal form can be written as

$$\dot{z} = Fz + G\xi$$
$$\dot{\xi} = A_r \xi + B_r [Hz + K\xi + bu]$$
$$y = C_r \xi.$$

It is also easy to check that the transfer function of the system can be expressed as

$$T(s) = b \frac{\det(sI - F)}{\det(sI - A)},$$

in which

$$A = \begin{pmatrix} F & G \\ B_r H & A_r + B_r K \end{pmatrix}.$$

From this it is concluded that in a (controllable and observable) linear system, the zeros of the transfer function $T(s)$ coincide with the eigenvalues of F. In other words, in a linear system the zero dynamics are linear dynamics whose eigenvalues coincide with the zeros of the transfer function of the system.

The Inverse System

Another property associated with the notion of zero of the transfer function, in a (single-input single-output) linear system, is the fact that the zeros characterize the dynamics of the inverse system (the latter being – loosely speaking – a system able to reproduce the input $u(t)$ from output $y(t)$ that this input has generated). This property has an immediate analogue for nonlinear systems. Considering system in normal form and setting

$$\mathbf{y}^{r-1}(t) = \mathrm{col}(y(t), y^{(1)}(t), \dots, y^{(r-1)}(t)),$$

it is easily seen that the input $u(t)$ can be determined as the output of a dynamical system, driven by $\mathbf{y}^{r-1}(t)$ and $y^{(r)}(t)$, modeled by

$$\begin{aligned} \dot{z} &= f_0(z, \mathbf{y}^{r-1}) \\ u &= \frac{y^{(r)} - q_0(z, \mathbf{y}^{r-1})}{b(z, \mathbf{y}^{r-1})}. \end{aligned} \qquad (1)$$

Thus, it is concluded that the unforced internal dynamics of the inverse system coincide with the zero dynamics as defined above.

It should be stressed, though, that the coincidence is limited to the case of single-input single-output systems. For a multi-input multi-output nonlinear systems, the link between zero dynamics and the dynamics of the inverse system is more subtle. This is essentially due to the fact that while the concept of zero dynamics only seeks to determine the dynamics compatible with

the constraint that the output is identically zero, the inverse system must describe all dynamics resulting in *any* admissible output function. As a consequence, computation of the zero dynamics and computation of the inverse system (whenever this is possible) are not equivalent and the latter is possible only under substantially stronger assumptions. The computation of the zero dynamics is based on an extension (Isidori 1995) of the classical algorithm of Wonham (1979) for the computation of the largest controlled invariant subspace in the kernel of the output map, while the computation of the inverse system is based on extensions, due to Hirschorn (1979) and Singh (1981) of the so-called structure algorithm introduced by Silverman (1969) for the computation of inverses and zero structure at the infinity. For a comparison of such assumptions and of their influence on the outcome of the associated algorithms, see Isidori and Moog (1988).

Input-Output Linearization

An appealing feature of the normal form described above is the straightforward observation that a (state) feedback law of the form

$$u = \frac{1}{b(z, \xi)}[-q_0(z, \xi) + K_r \xi + v]$$

changes the system into a system

$$\begin{aligned} \dot{z} &= f_0(z, \xi) \\ \dot{\xi} &= (A_r + B_r K_r)\xi + B_r v \\ y &= C_r \xi \end{aligned}$$

whose input-output behavior (between input v and output y) is *fully linear* (and stable if K_r is chosen so that the matrix $A_r + B_r K_r$ in Hurwitz). In fact, the law in question renders the system partially unobservable, with all nonlinearities confined to its unobservable part (Isidori et al. 1981). This control law is clearly non-robust, as it relies upon exact cancelation of possibly uncertain terms, but it can be rendered robust by means of appropriate dynamic compensation (Freidovich and Khalil 2008).

The system obtained in this way has the structure of a cascade of two sub-systems, one of which, modeled as

$$\dot{z} = f_0(z, \xi),$$

is seen as "driven" by the input ξ. This motivates the interest in classifying the asymptotic properties of such subsystem, as discussed below.

Asymptotic Properties of the Zero Dynamics

Linear systems with no zeroes in the right-half complex plane are traditionally called *minimum-phase* systems, in view of certain properties of the Bode gain and phase plots of its transfer function. Thus, in view of the interpretation given above, linear systems whose zero dynamics are asymptotically stable are minimum-phase systems. This terminology has been (somewhat abusively, but with the clear intent of providing a concise and expressive characterization) borrowed to classify nonlinear systems whose zero dynamics have desirable (from the stability viewpoint) properties. Assuming that $z = 0$ is an equilibrium of $\dot{z} = f_0(z, 0)$, the following cases are considered:

- A nonlinear system is *locally minimum-phase* (respectively, *locally exponentially minimum-phase*) if the equilibrium $z = 0$ of $\dot{z} = f_0(z, 0)$ is locally asymptotically (respectively locally exponentially) stable (Byrnes and Isidori 1984).
- A nonlinear system is *globally minimum-phase* if the equilibrium $z = 0$ of $\dot{z} = f_0(z, 0)$ is globally asymptotically stable (Byrnes and Isidori 1991).
- A nonlinear system is *strongly minimum-phase* if the system $\dot{z} = f_0(z, \xi)$, viewed as a system with input ξ and state z, is input-to-state stable (Liberzon 2002).

According to the well-known criterion of Sontag (1995) for input-to-state stability, a system is strongly minimum phase if and only if there exists a positive definite and proper smooth real-valued function $V(z)$, class $\mathcal{K}_{\infty}$

functions $\underline{\alpha}(\cdot), \overline{\alpha}(\cdot), \alpha(\cdot)$ and a class $\mathcal{K}$ function $\chi(\cdot)$ satisfying

$$\underline{\alpha}(|z|) \leq V(z) \leq \overline{\alpha}(|z|) \qquad \forall z$$
$$\frac{\partial V}{\partial z} f_0(z, \xi) \leq -\alpha(|z|) \quad \forall (z, \xi)$$
$$\text{such that } |z| \geq \chi(|\xi|).$$

As a special case, it is seen that a system is globally minimum phase if and only if there exists a function $V(z)$, bounded as above, such that

$$\frac{\partial V}{\partial z} f_0(z, 0) \leq -\alpha(|z|) \qquad \forall z.$$

If, instead, the weaker inequality

$$\frac{\partial V}{\partial z} f_0(z, 0) \leq 0 \qquad \forall z$$

holds, the system is said to be *globally weakly minimum-phase*.

The criterion summarized above is of paramount importance in the design of feedback laws to the purpose of stabilizing nonlinear systems that are globally (or strongly) minimum phase, as it will be seen below.

Zero Dynamics and Stabilization

The first and foremost immediate implication of the properties described above is the fact that the feedback law

$$u = \frac{1}{b(z, \xi)}[-q_0(z, \xi) + K_r \xi],$$

if K_r is chosen so that the matrix $A_r + B_r K_r$ in Hurwitz, globally asymptotically stabilizes the equilibrium $(z, \xi) = (0, 0)$ of a strongly minimum-phase system. In fact, as observed, the corresponding closed-loop system can be seen as an asymptotically stable (linear) system driving an input-to-state stable (nonlinear) system. As already observed, this control mode is non-robust (as it relies upon exact cancelations) and requires the availability of the full state (z, ξ) of the controlled system. However, both these

deficiencies can be to some extent fixed, by means of appropriate techniques, that will be briefly reviewed below.

If the requirement of global stability is replaced by the (weaker) requirement of *stability with a guaranteed region of attraction*, then the desired control goal can be achieved by means of a much simpler law, depending only on the partial state ξ and not requiring cancelations. Stability with a guaranteed region of attraction essentially means that a given equilibrium is rendered asymptotically stable, with a region of attraction that contains an a priori *fixed* compact set. In this context, the most relevant results can be summarized as follows.

Assume the system possesses a globally defined normal form and, without loss of generality, let $b(z, \xi) > 0$. Let the system be controlled by a "partial state" feedback of the form

$$u = -kK_r\xi\,,$$

in which $k \in \mathbb{R}$. Under this control mode, the following results are obtained:

- Suppose the system is strongly minimum phase. Then, there is a matrix K_r and, for every choice of a compact set $\mathcal{C}$ and of a number $\varepsilon > 0$, there are a number k^* and a time T^* such that, if $k \geq k^*$, all trajectories of the closed-loop system with initial condition in $\mathcal{C}$ are bounded and satisfy $|x(t)| \leq \varepsilon$ for all $t \geq T^*$.

- Suppose the system is strongly minimum phase and also locally exponentially minimum phase. Suppose $q_0(0, 0) = 0$. Then, there is a matrix K_r and, for every choice of a compact set $\mathcal{C}$ there is a number k^* such that, if $k \geq k^*$, the equilibrium $x = 0$ of the system is locally asymptotically stable, with a domain of attraction that contains the set $\mathcal{C}$.

In these results, the system is stabilized by means of a *static* control law that depends only on the partial state ξ and not on the (possibly unknown) quantities $q_0(z, \xi)$, $b(z, \xi)$. Bearing in mind the fact that the r components of ξ coincide with the output y and its derivatives $y^{(1)}, \ldots, y^{(r-1)}$, it is possible to replace the

control in question by means of a *dynamic* control law that only depends on the output y, following a design paradigm originally proposed by H. Khalil. In fact, if the system is strongly minimum phase and also locally exponentially minimum phase and if $q_0(0, 0) = 0$, asymptotic stability with a guaranteed region of attraction can be achieved by means of dynamical feedback law of the form Khalil and Esfandiari (1993)

$$\begin{aligned}
\dot{\hat{\xi}}_1 &= \hat{\xi}_2 + \kappa c_{r-1}(y - \hat{\xi}_1)\\
\dot{\hat{\xi}}_2 &= \hat{\xi}_3 + \kappa^2 c_{r-2}(y - \hat{\xi}_1)\\
&\quad\cdots\\
\dot{\hat{\xi}}_{r-1} &= \hat{\xi}_r + \kappa^{r-1} c_1(y - \hat{\xi}_1)\\
\dot{\hat{\xi}}_r &= \kappa^r c_0(y - \hat{\xi}_1)\\
u &= -\sigma_L(kK_r\hat{\xi})\,,
\end{aligned}$$

in which κ and the c_i are design parameters and $\sigma_L(s)$ is a smooth saturation function, characterized as follows: $\sigma_L(s) = s$ if $|s| \leq L$, $\sigma_L(s)$ is odd and monotonically increasing, with $0 < \sigma_L'(s) \leq 1$, and $\lim_{s\to\infty} \sigma_L(s) = L(1 + c)$ with $0 < c \ll 1$. The number L is a design parameter also.

It is also possible to show that a suitable "extension" of this dynamic feedback law can be used to asymptotically recover the effects of the input-output linearizing law considered earlier. In this way, the lack of robustness intrinsically present in such control law is overcome (Freidovich and Khalil (2008)).

Output Regulation

The concept of zero dynamics plays a fundamental role in the problem of output regulation. The problem in question considers a controlled plant modeled by

$$\begin{aligned}
\dot{x} &= f(w, x, u)\\
e &= h(w, x)\,,
\end{aligned}$$

in which u is the control input, w is a set of exogenous variables (command and disturbances),

and e is a set of regulated variables. The exogenous variables are thought of as generated by an autonomous system

$$\dot{w} = s(w)$$

known as the *exosysten*. The problem is to design a (possibly dynamic) controller

$$\dot{x}_c = f_c(x_c, e)$$
$$u = h_c(x_c, e)$$

driven by the regulated variable e, such that in the resulting closed-loop system all trajectories are ultimately bounded and $\lim_{t \to \infty} e(t) = 0$. The problem in question has been the object of intensive research in the past years. In what follows we limit ourselves to highlight the role of the concept of zero dynamics in this problem.

Assume that the set W where the exosystem evolves is compact and invariant and suppose a controller exists that solves the problem of output regulation. Then, the associated closed-loop has a steady-state locus (see Isidori and Byrnes 2008), the graph of a possibly set-valued map defined on W. Suppose the map in question is single-valued, which means that for each given exogenous input function $w(t)$, there exists a *unique* steady-state response, expressed as $x(t) = \pi(w(t))$ and $x_c(t) = \pi_c(w(t))$. If, in addition, $\pi(w)$ and $\pi_c(w)$ are continuously differentiable, it is readily seen that

$$L_s \pi(w) = f(w, \pi(w), \psi(w))$$
$$0 = h(w, \pi(w))$$
$$L_s \pi_c(w) = f_c(\pi_c(w), 0) \qquad \forall w \in W.$$
$$\psi(w) = h_c(\pi_c(w), 0)$$

The first two equations, introduced in Isidori and Byrnes (1990), are known as the *nonlinear regulator equations*. They clearly show that the graph of the map $\pi(w)$ is a manifold contained in the zero set of the output map e, rendered invariant by the control $u = \psi(w)$. In particular, the steady-state trajectories of the closed-loop system are trajectories of the *zero dynamics* of

the controlled plant. The second two equations, on the other hand, interpret the ability, of the controller, to generate the feedforward input necessary to keep $e(t) = 0$ in steady-state. This is a nonlinear version of the well-known *internal model principle* of Francis and Wonham (1975).

Passivity

Consider a nonlinear input-affine system having the same number m of inputs and outputs and recall that this system is said to be *passive* if there exists a continuous nonnegative function real-valued function $W(x)$, with $W(0) = 0$, that satisfies

$$W(x(t)) - W(x(0)) \leq \int_0^t y^T(s)u(s)ds$$

along trajectories. The function $W(x)$ is the so-called *storage function* of the system.

It is well known that the notion of passivity plays an important role in system analysis and that the theory of passive systems leads to powerful methodologies for the design of feedback laws for nonlinear systems. In this context, the question of whether a given, non-passive, nonlinear system could be rendered passive by means of state feedback is indeed relevant. It turns out that this possibility can be simply expressed as a property of the zero dynamics of the system.

Suppose that $L_g h(x)$ is nonsingular and set $g^*(x) = g(x)[L_g h(x)]^{-1}$. If the m columns of $g^*(x)$ are complete and commuting vector fields, there exists a globally defined change of coordinates that brings the system in normal form

$$\dot{z} = f_0(z, y)$$
$$\dot{y} = q_0(z, y) + b(z, y)u$$

Then, there exists a feedback law $u = \alpha(z, y)$ that renders the resulting closed-loop system passive, with a C^2 and positive definite storage function $W(x)$, if and only if the system is globally weakly minimum phase (Byrnes et al. 1991).

Limits of Performance

It is well-known that linear systems having zeros in the left-half plane are difficult to control, and obstruction exists to the fulfillment of certain control specifications. One of these is found in the analysis of the so-called *cheap control problem*, namely, the problem of finding a stabilizing feedback control that minimizes the functional

$$J_\varepsilon = \frac{1}{2} \int_0^\infty [y^{\mathrm{T}}(t)y(t) + \varepsilon u^{\mathrm{T}}(t)u(t)]dt$$

when $\varepsilon > 0$ is small. As $\varepsilon \to 0$, the optimal value J_ε^* tends to J_0^*, the *ideal performance*. It is well-known that, in a linear system, $J_0^* = 0$ if and only if the system is minimum phase and right invertible and, in case the system has zeros with positive real part, it is possible to express explicitly J_0^* in terms of the zeros in question. If the (linear) system is expressed in normal form as

$$\dot{z} = Fz + G\xi$$
$$\dot{\xi} = Hz + K\xi + bu$$
$$y = \xi$$

with $b \neq 0$, and the zero dynamics are antistable (that is *all* the eigenvalues of F have positive real part), it can be shown that J_0^* coincides with the minimal value of the energy

$$J = \frac{1}{2} \int_0^\infty \xi^{\mathrm{T}}(t)\xi(t)dt$$

required to stabilize the (antistable) system $\dot{z} = Fz + G\xi$. In other words, the limit as $\varepsilon \to 0$ of the optimal value of J_ε is equal to the least amount of energy required to stabilize the dynamics of the inverse system.

This result has an appealing nonlinear counterpart (Seron 1999). In fact, for a nonlinear input-affine system having the same number m of inputs and outputs in normal form, with $f_0(z, \xi)$ of the form $f_0(z, \xi) = f_0(z) + g_0(z)\xi$ and $\dot{z} = f_0(z)$ antistable, under appropriate technical assumptions (mostly related to the existence of the solution of the associated optimal control problems), the same result holds: the lowest attainable

value of the L_2 norm of the output coincides with the least amount of energy required to stabilize the dynamics of z.

Summary and Future Directions

The concept of zero dynamics plays an important role in a large number of problems arising in analysis and design of nonlinear control systems, among which the most relevant ones are the problems of asymptotic stabilization and those of asymptotic tracking/rejection of exogenous command/disturbance inputs. Essentially, all such applications deal with single-input single-output systems, require the system to be preliminarily reduced to a special form by means of appropriate change of coordinates, and assume the dynamics in question to be globally asymptotically stable. The analysis of systems having many inputs and many outputs, of systems in which normal forms cannot be defined, and of systems in which the zero dynamics are unstable is still a challenging and unexplored area of research.

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Input-to-State Stability
▶ Regulation and Tracking of Nonlinear Systems

Bibliography

Byrnes CI, Isidori A (1984) A frequency domain philosophy for nonlinear systems. IEEE Conf Dec Control 23:1569–1573

Byrnes CI, Isidori A (1991) Asymptotic stabilization of minimum-phase nonlinear systems. IEEE Trans Autom Control AC-36:1122–1137

Byrnes CI, Isidori A, Willems JC (1991) Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear aystems. IEEE Trans Autom Control AC-36:1228–1240

Francis BA, Wonham WM (1975) The internal model principle for linear multivariable regulators. J Appl Math Optim 2:170–194

Freidovich LB, Khalil HK (2008) Performance recovery of feedback-linearization-based designs. IEEE Trans Autom Control 53:2324–2334

Hirschorn RM (1979) Invertibility for multivariable non-linear control systems. IEEE Trans Autom Control AC-24:855–865

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York

Isidori A, Byrnes CI (1990) Output regulation of nonlinear systems. IEEE Trans Autom Control AC-35:131–140

Isidori A, Byrnes CI (2008) Steady-state behaviors in nonlinear systems, with an application to robust disturbance rejection. Ann Rev Control 32:1–16

Isidori A, Moog C (1988) On the nonlinear equivalent of the notion of transmission zeros. In: CI Byrnes, A Kurzhanski (eds) Modelling and adaptive control. Lecture notes in control and information sciences, vol 105. Springer, Berlin/New York pp 445–471

Isidori A, Krener AJ, Gori-Giorgi C, Monaco S (1981) Nonlinear decoupling via feedback: a differential geometric approach. IEEE Trans Autom Control AC-26:331–345

Khalil HK, Esfandiari F (1993) Semiglobal stabilization of a class of nonlinear systems using output feedback. IEEE Trans Autom Control AC-38:1412–1415

Liberzon D, Morse AS, Sontag ED (2002) Output-input stability and minimum-phase nonlinear systems. IEEE Trans Autom Control AC-43:422–436

Seron MM, Braslavsky JH, Kokotovic PV, Mayne DQ (1999) Feedback limitations in nonlinear systems: from Bode integrals to cheap control. IEEE Trans Autom Control AC-44:829–833

Singh SN (1981) A modified algorithm for invertibility in nonlinear systems. IEEE Trans Autom Control AC-26:595–598

Silverman LM (1969) Inversion of multivariable linear systems. IEEE Trans Autom Control AC-14:270–276

Sontag ED (1995) On the input–to–state stability property. Eur J Control 1:24–36

Wonham WM (1979) Linear multivariable control: a geometric approach. Springer, New York

Nonparametric Techniques in System Identification

Rik Pintelon and Johan Schoukens
Department ELEC, Vrije Universiteit Brussel, Brussels, Belgium

Abstract

This entry gives an overview of classical and state-of-the-art nonparametric time and frequency-domain techniques. In opposition to parametric methods, these techniques require no detailed structural information to get insight into the dynamic behavior of complex systems. Therefore, nonparametric methods are used in system identification to get an initial idea of the model complexity and for model validation purposes (e.g., detection of unmodeled dynamics). Their drawback is the increased variability compared with the parametric estimates. Although the main focus of this entry is on the classical identification framework (estimation of dynamical systems operating in open loop from known input, noisy output observations), the reader will also learn more about (i) the connection between transient and leakage errors, (ii) the estimation of dynamical systems operating in closed loop, (iii) the estimation in the presence of input noise, and (iv) the influence of nonlinear distortions on the linear framework. All results are valid for discrete- and continuous-time systems. The entry concludes with some user choices and practical guidelines for setting up a system identification experiment and choosing an appropriate estimation method.

Keywords

Best linear approximation; Correlation method; Empirical transfer function estimate; Errors-in-variables; Feedback; Frequency response function; Gaussian process regression; Impulse transient response modeling method; Local polynomial method; Local rational method; Noise (co)variances; Noise power spectrum; Spectral analysis

Introduction

Nonparametric representations such as frequency response functions (FRFs) and noise power spectra are very useful in system identification: they are used (i) to verify the quality of the identification experiment (high or poor signal-to-noise ratio?), (ii) to get quickly insight into the dynamic behavior of the plant (complex or easy identification problem?), and (iii) to validate the

parametric plant and noise models (detection of unmodeled dynamics); see also ▶ System Identification: An Overview. In addition, via specially designed periodic excitation signals, it is possible to detect and quantify the nonlinear distortions in the FRF estimate. As such, without estimating a parametric model, the users can easily decide whether or not the linear framework is accurate enough for their particular application.

The estimation of the nonparametric models typically starts from sampled input-output signals $u(nT_s)$ and $y(nT_s), n = 0, 1, \ldots, N - 1$, that are transformed to the frequency domain via the discrete Fourier transform (DFT)

$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(nT_s) e^{-j2\pi kn/N} \quad (1)$$

with T_s the sampling period, $x = u$ or y, and $X = U$ or Y. One of the main difficulties in estimating an FRF and noise power spectrum is the leakage error in the DFT spectrum $X(k) = \text{DFT}(x(t))$ (1). It is due to the finite duration NT_s of the experiment, and it increases the mean square error of the nonparametric estimates. Therefore, all methods try to suppress the leakage error as much as possible.

This entry starts by a detailed analysis of the leakage problem (section "The Leakage Problem"), followed by an overview of standard and advanced nonparametric time (section "Nonparametric Time-Domain Techniques") and frequency (section "Nonparametric Frequency-Domain Techniques") domain techniques. First, it is assumed that the system operates in open loop (see Fig. 1) and that known input, noisy output observations are available (sections "Nonparametric Time-Domain Techniques" and "Nonparametric Frequency-Domain Techniques"). Next, section "Extensions" extends the results to systems operating in closed loop (section "Systems Operating in Feedback"); to noisy input, noisy output observations (section "Noisy Input, Noisy Output Observations"); and to nonlinear systems (section "Nonlinear Systems"). Finally, some user choices are discussed (section "User Choices") and
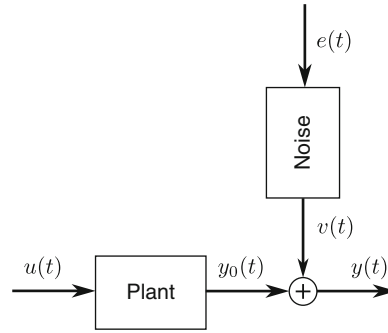
Nonparametric Techniques in System Identification, Fig. 1 Classical identification framework: discrete- or continuous-time plant operating in open loop; known input $u(t)$, noisy output $y(t)$ observations; and $v(t)$ filtered discrete-time or band-limited continuous-time white noise $e(t)$ that is independent of $u(t)$. $y_0(t)$ denotes the true output of the plant. In the continuous-time case, it is assumed that the unobserved driving noise source $e(t)$ has finite variance and constant (*white*) power spectrum within the acquisition bandwidth

some practical guidelines are given (section "Guidelines"). Unless otherwise stated, the input $u(t)$ and the disturbing noise $v(t)$ are assumed to be statistically uncorrelated.

The Leakage Problem

For arbitrary excitations $u(t)$, the relationship between the true input $U(k)$ and true output $Y_0(k)$ DFT spectra (1) of a linear dynamic system is given by

$$Y_0(k) = G(\Omega_k)U(k) + T_G(\Omega_k) \quad (2)$$

where $\Omega_k = j\omega_k$ or $\exp(-j\omega_k T_s)$ for, respectively, continuous- and discrete-time systems; $\omega_k = 2\pi k/(NT_s)$; $G(\Omega_k)$ the plant frequency response function; and $T_G(\Omega_k)$ the leakage error due to the plant dynamics (Pintelon and Schoukens 2012, Section 6.3.2). The leakage error $T_G(\Omega)$ is a smooth function of the frequency that decreases to zero as $O(N^{-1/2})$ for N increasing to infinity. It depends on the difference between the initial and final conditions of the experiment and has exactly the same poles as the plant transfer function. Therefore, the

time-domain response of $T_G(\Omega)$ is decaying exponentially to zero as a transient error.

From this short discussion, it can be concluded that the leakage error in the frequency domain is equivalent to the transient error in the time domain. The only difference being that the former depends on the difference between the initial and final conditions, while the latter solely depends on the initial conditions.

Standard spectral analysis methods (see section "Spectral Analysis Method") suppress the leakage term $T_G(\Omega_k)$ in (2) by multiplying the time-domain signals with a window $w(t)$ before taking the DFT (1)

$$(X(k))W = \frac{1}{\sqrt{N}\,w_{\mathrm{rms}}} \sum_{n=0}^{N-1} w(nT_s)x(nT_s)e^{-j2\pi\frac{kn}{N}} \tag{3}$$

with $w_{\mathrm{rms}} = \left(\sum_{n=0}^{N-1} |w(nT_s)|^2 / N\right)^{1/2}$ the root mean square (rms) value of the window $w(t)$. The scaling in (3) is such that the transformation preserves the rms value of the signal. The relationship between the DFT spectra $(U(k))_W$ and $(Y_0(k))_W$ of the windowed input-output signals $w(t)u(t)$ and $w(t)y_0(t)$ is given by

$$(Y_0(k))W = G(\Omega_k)\,(U(k))_W + E_{\mathrm{int}}(k) + E_{\mathrm{leak}}(k) \tag{4}$$

where $E_{\mathrm{int}}(k)$ and $E_{\mathrm{leak}}(k)$ are, respectively, the interpolation error and the remaining leakage error

$$E_{\mathrm{int}}(k) = (G(\Omega_k)U(k))_W - G(\Omega_k)\,(U(k))_W \tag{5}$$

$$E_{\mathrm{leak}}(k) = (T_G(\Omega_k))_W \tag{6}$$

Note that $E_{\mathrm{int}}(k) = 0$ if $G(\Omega_k)$ is constant within the bandwidth of $W(k)$, while the interpolation error is large if the FRF varies significantly within the window bandwidth. To keep $E_{\mathrm{int}}(k)$ small, the frequency resolution $1/(NT_s)$ should be sufficiently large and the window bandwidth should be small enough. On the other hand, a larger window bandwidth is beneficial for reducing the leakage error $E_{\mathrm{leak}}(k)$. Hence, choosing an appropriate window for nonparametric FRF

and noise power spectrum estimation is making a trade-off between the reduction of the leakage error $E_{\mathrm{leak}}(k)$ and the increase of the interpolation error $E_{\mathrm{int}}(k)$ (Schoukens et al. 2006).

Note that exactly the same analysis can be made for the continuous- or discrete-time dynamics of the disturbing output noise $v(t)$ in Fig. 1

$$V(k) = H(\Omega_k)E(k) + T_H(\Omega_k) \tag{7}$$

with $H(\Omega_k)$ the noise frequency response function, $E(k)$ the DFT of the unobserved driving discrete-time or band-limited continuous-time white noise source $e(t)$ (Pintelon and Schoukens 2012, Section 6.7.3), and $T_H(\Omega_k)$ the noise leakage (transient) term. The noise leakage term is often neglected but can be important for lightly damped systems (e.g., in modal analysis). Most nonparametric techniques suppress the sum of the plant and noise leakage errors $T_G(\Omega_k) + T_H(\Omega_k)$.

If an integer number of periods of the steady-state response to a periodic excitation is measured, then the plant leakage error $T_G(\Omega_k)$ in (2) is zero, which simplifies significantly the estimation problem. Therefore, for the frequency-domain techniques, a distinction is made between periodic and nonperiodic excitations. Note, however, that the noise leakage (transient) term $T_H(\Omega_k)$ in (7) remains different from zero.

Nonparametric Time-Domain Techniques

The time-domain methods estimate the impulse response of the plant via the time-domain relationship that the true output $y_0(t)$ equals the convolution product between the impulse response $g(t)$ and the true input $u(t)$. For discrete-time systems, it takes the form

$$y_0(t) = \sum_{n=0}^{\infty} g(n)u(t-n) \tag{8}$$

In practice only a finite number of impulse response coefficients $g(t)$ can be estimated from

N input-output samples, and, therefore, (8) is approximated by a finite sum

$$y_0(t) \approx \sum_{n=0}^{L} g(n)u(t-n) \qquad (9)$$

where $L \leq N - 1$ should also be determined from the data. From (9), it can be seen that the response depends on the past input values $u(-1), u(-2), \ldots, u(-L)$. Since these values are unknown, an exponentially decaying transient error is present in the first L samples of the predicted output (9). This transient error is the time-domain equivalent of the leakage error $T_G(\Omega_k)$ in (2). To remove the transient error, the first L output samples can be discarded in the predicted output (9). It reduces the amount of data from N to $N - L$ and, hence, increases the mean square error of the estimates. If it is known that the transfer function has no direct term, then $g(0) = 0$, and the sum (9) starts from $n = 1$.

Correlation Methods

Correlation methods have been studied intensively since the end of the 1950s (see Eykhoff 1974) and are nowadays still used in telecommunication channel estimation and equalization. The impulse response coefficients are found by minimizing the sum of the squared differences between the observed output samples and the output samples predicted by (9)

$$\sum_{t=L}^{N-1} \left(y(t) - \sum_{n=0}^{L} g(n)\, u(t-n) \right)^2 \qquad (10)$$

w.r.t. $g(m), m = 0, 1, \ldots, L$. The solution of this linear least squares problem is given by the famous Wiener-Hopf equation

$$\hat{R}_{yu}(m) = \sum_{n=0}^{L} g(n)\, \hat{R}_{uu}(m-n) \qquad (11)$$

for $m = 0, 1, \ldots, L$, where $\hat{R}_{yu}$ and $\hat{R}_{uu}$ are estimates of, respectively, the cross- and autocorrelation functions $R_{yu}(\tau) = \mathbb{E}\{y(t)u(t-\tau)\}$ and $R_{uu}(\tau) = \mathbb{E}\{u(t)u(t-\tau)\}$

$$\hat{R}_{yu}(m) = \frac{1}{N-L} \sum_{t=L}^{N-1} y(t)u(t-m) \qquad (12)$$

$$\hat{R}_{uu}(m-n) = \frac{1}{N-L} \sum_{t=L}^{N-1} u(t-n)u(t-m) \qquad (13)$$

(Godfrey 1993, Chapter 1; Ljung 1999, Chapter 6). Since the number of estimated impulse response coefficients L can grow with the amount of data N, the correlation method (11) is classified as being nonparametric. If the input is white noise, then the expected value of $\hat{R}_{uu}(m)$ is proportional to the Kronecker delta $\delta(m)$, and the cross-correlation $\hat{R}_{yu}(m)$ (11) is – within a scaling factor – a good approximation of the impulse response. This property is used in blind channel estimation.

Gaussian Process Regression

The linear least squares (10) solution can be (very) sensitive to disturbing output noise if L is not much smaller than N. This problem is circumvented by the Gaussian process regression approach. The key idea consists in modeling the impulse response coefficients $g(n)$ as a zero-mean Gaussian process with a certain covariance structure P_L that depends on a few hyper-parameters (Pillonetto et al. 2011). In Chen et al. (2012), it has been shown that the Gaussian process regression is equivalent to the following regularized (see also ▶ System Identification Techniques: Convexification, Regularization, and Relaxation) linear least squares problem

$$\sum_{t=L}^{N-1} \left(y(t) - \sum_{n=0}^{L} g(n)u(t-n) \right)^2 + \sigma^2 g^T P_L^{-1} g \qquad (14)$$

where $g = (g(0), g(1), \ldots, g(L))^T$ and with σ^2 the variance of the output disturbance. The hyper-parameters defining P_L and the noise variance σ^2 are estimated via an empirical Bayes method.

Nonparametric Frequency-Domain Techniques

The frequency-domain techniques estimate the frequency response function (FRF) using relationship (2) or (4) between the input-output DFT spectra. We start with the simplest approach and gradually increase the complexity of the estimation methods. Note that nonparametric FRF estimation is still a quickly evolving research area, such that the pros and cons of the advanced methods are yet not well established.

Empirical Transfer Function Estimation

If an integer number of periods P of the steady-state response to a *periodic excitation* is observed, then the leakage term in $T_G(\Omega_k)$ in (2) is zero, and the FRF is estimated by dividing the output by the input DFT spectra at the *excited frequencies* (Pintelon and Schoukens 2012, Section 2.4)

$$\hat{G}(\Omega_k) = \frac{Y(k)}{U(k)} \qquad (15)$$

The output noise variance $\sigma_V^2(k)$ is estimated via the sample variance $\hat{\sigma}_V^2(k)$ of the output DFT spectra over the P consecutive signal periods. The variance of the FRF estimate (15) is then given by

$$\text{var}(\hat{G}(\Omega_k)) = \frac{\sigma_V^2(k)}{P\,|U(k)|^2} \qquad (16)$$

where $|U(k)|$ is the magnitude of $U(k)$.

Applying (15) to *random excitations* gives the empirical transfer function estimate (Ljung 1999, Section 6.3). Due to the presence of the plant leakage error $T_G(\Omega_k)/U(k)$, the statistical properties of (15) for random inputs are quite different from those for periodic inputs. While the empirical transfer function estimate (ETFE) is unbiased and has finite variance (16) for periodic inputs, it is biased and has infinite variance for random inputs (Broersen 2004). To improve the statistical properties of the ETFE for random inputs, one can either approximate locally the ETFE by a polynomial (Stenman et al. 2000) or perform a weighted average of ETFEs over

subrecords of the total response (Ljung 1999, Section 6.4). In Heath (2007), it is shown that the optimally (in mean square sense) weighted ETFE equals the spectral analysis method.

Spectral Analysis Method

The spectral analysis method is available in any digital spectrum analyzer. It is based on the relationship between the FRF and the cross- and autopower spectra of the input-output signals

$$G(\Omega) = \frac{S_{yu}(\Omega)}{S_{uu}(\Omega)} = \frac{F\{R_{yu}(\tau)\}}{F\{R_{uu}(\tau)\}} \qquad (17)$$

with $F\{\}$ the Fourier transform (Bendat and Piersol 1980, Chapter 4; Brillinger 1981, Chapter 8). Comparing (11) and (17), it can be seen that the spectral analysis method is the frequency-domain equivalent of the correlation method (take the Fourier transform of the expected value of (11)). There are basically two methods for estimating the cross- and autopower spectra in (17) from sampled data: the Blackman and Tukey (1958) and the Welch (1967) procedures.

The Blackman-Tukey procedure (Blackman and Tukey 1958; Ljung 1999, Section 6.4) consists in taking the DFT (3) of the windowed cross- and autocorrelation functions, viz.,

$$\hat{R}_{yu}(\tau) = \frac{1}{N} \sum_{t=}^{N-1} y(t)\, u(t - \tau) \qquad (18)$$

$$\hat{S}_{Ryu}(k) = \frac{1}{\sqrt{N}} \sum_{\tau=0}^{N-1} w(\tau)\, \hat{R}_{yu}(\tau) e^{-j2\pi \frac{k\tau}{N}} \quad (19)$$

resulting in an FRF estimate (17) at the full frequency resolution $1/(NT_s)$ of the measurement. It can be shown that (19) is a smoothed version of the periodogram $Y(k)\overline{U(k)}$, where is $\bar{U}$ the complex conjugate of U (Brillinger 1981, Chapter 5).

In the Welch approach (Welch 1967; Pintelon and Schoukens 2012, Section 2.6), the N input-output samples are split into M subrecords of N/M samples each, and the DFT spectra $(U^{[m]}(k))_W$ and $(Y^{[m]}(k))_W$ of the windowed

input and output samples are calculated via (3) where N is replaced by N/M, giving

$$\hat{S}_{Y_W U_W}(k) = \frac{1}{M} \sum_{m=1}^{M} \left(Y^{[m]}(k)\right) W \overline{\left(U^{[m]}(k)\right) W} \tag{20}$$

$$\hat{S}_{U_W U_W}(k) = \frac{1}{M} \sum_{m=1}^{M} \left|\left(U^{[m]}(k)\right) W\right|^2 \tag{21}$$

The spectral analysis estimate of the FRF and its variance are then given by

$$\hat{G}(\Omega_k) = \frac{\hat{S}_{Y_W U_W}(k)}{\hat{S}_{U_W U_W}(k)} \tag{22}$$

$$\mathrm{var}(\hat{G}(\Omega_k)) \approx \frac{\sigma_V^2(k)}{M} \mathbb{E}\left\{\hat{S}_{U_W U_W}^{-1}(k)\right\} \tag{23}$$

(Brillinger 1981, Chapter 8; Heath 2007). Finally, the output noise variance $\sigma_V^2(k)$ in (23) is estimated as

$$\hat{\sigma}_V^2(k) = \frac{M}{M-1}\left(\hat{S}_{Y_W Y_W}(k) - \frac{\left|\hat{S}_{Y_W U_W}(k)\right|^2}{\hat{S}_{U_W U_W}(k)}\right) \tag{24}$$

(Brillinger 1981, Chapter 8; Pintelon and Schoukens 2012, Section 2.5.4). Due to the spectral width of the window used, the estimates (22) and (24) are correlated over the frequency (the correlation length is about twice the spectral width). Note that (21) is used for estimating noise power spectra (Brillinger 1981, Chapter 5). Note also that for *periodic excitations* combined with a rectangular window $w(nT_s) = 1$, the spectral analysis estimate (22), where each subrecord is equal to a signal period, simplifies to the ETFE (15).

Compared with the Blackman-Tukey procedure (19), the FRF estimate (22) based on the Welch approach (20) and (21) has a frequency resolution and a variance (23) that are M times smaller. In measurement devices, the FRFs are estimated using the Welch approach (20)–(22) where each subrecord is an independent measurement with a fixed number of samples. The reason for this is that the cross- and autopower spectra estimates (20) and (21) can easily be updated as

more experiments (input-output data records) are available. If the number of measured records M increases to infinity, then (22) converges to the true value, provided a perfect suppression of the leakage error.

In measurement devices, the quality of the spectral analysis estimate (22) is often quantified via the coherence $\gamma^2(\omega)$

$$\gamma^2(\omega) = \frac{\left|S_{yu}(\Omega)\right|^2}{S_{yy}(\Omega) S_{uu}(\Omega)} \tag{25}$$

which is comprised between 0 and 1. It is related to the variance of the spectral analysis estimate as

$$\mathrm{var}(\hat{G}(\Omega_k)) = \frac{1 - \gamma^2(\omega_k)^2}{\gamma}(\omega_k) \left|G(\Omega_k)\right|^2$$

A coherence smaller than 1 indicates the presence of disturbing noise, residual leakage errors, nonlinear distortions, or a nonobserved input.

Following the same lines of Welch (1967), the statistical properties of the spectral analysis estimate (22) can be improved via overlapping subrecords in the cross- and autopower spectra estimates (20) and (21). This has been studied in detail for noise power spectra in Carter and Nuttall (1980) and for FRFs in Antoni and Schoukens (2007).

Advanced Methods

The goal of the advanced methods is to estimate the FRF at the full frequency resolution $1/(NT_s)$ of the experiment duration NT_s while suppressing the influence of the leakage and the noise errors. Without some extra information, it is impossible to achieve this goal via (2). The additional piece of information that allows one to solve the problem is that the FRF and the leakage error are locally smooth functions of the frequency.

The *local polynomial method* (Pintelon and Schoukens 2012, Chapter 7) approximates the FRF and the leakage error in (2) locally in the frequency band $[k - n, k + n]$ by a polynomial. From the residuals of the local linear least squares solution, one also gets an estimate of the output noise variance σ_V^2 and, hence, also of the variance of the FRF. The whole procedure is repeated for

all DFT frequencies k in the frequency band of interest. The correlation length of the estimates equals $\pm 2n$, which is twice the local bandwidth of the polynomial approximation.

The *local rational method* (McKelvey and Guérin 2012) follows the same lines as the local polynomial method, except that the FRF and the leakage error in (2) are locally approximated by rational forms with the same poles ($G = B/A$ and $T_G = I/A$). Due to the common poles, the local rational approximation problem can be transformed into a local linear least squares problem. The method is biased but suppresses better the plant leakage error of lowly damped systems.

The *transient impulse response modeling method* (Hägg and Hjalmarsson 2012) approximates the FRF and the leakage error by, respectively, finite impulse and transient response models, giving a large sparse global linear least squares problem. From the residuals of the global linear least squares solution, one gets an estimate of the output noise variance σ_V^2 and, hence, also of the variance of the FRF. This approach has the best smoothing properties and is recommended in case the noise error is dominant.

Extensions

In sections "Nonparametric Time-Domain Techniques" and "Nonparametric Frequency-Domain Techniques," it is assumed that the linear plant operates in open loop and that the input is known exactly. If the plant operates in feedback and/or the input observations are noisy, then the presented time and frequency-domain techniques are biased. In sections "Systems Operating in Feed-

back" and "Noisy Input, Noisy Output Observations," it is shown that the estimation bias can be avoided if a known external reference signal is available (typically the signal stored in the arbitrary waveform generator).

Since most real-life systems behave to some extent nonlinearly, it is important to detect and quantify the nonlinear effects in FRF estimates. This issue is handled in section "Nonlinear Systems."

Systems Operating in Feedback

The key difficulty of estimating the FRF of a plant operating in feedback (see Fig. 2) using *nonperiodic excitations* is that the true input $u(t)$ is correlated with the process noise $v(t)$. The direct approaches of sections "Nonparametric Time–Domain Techniques" and "Nonparametric Frequency-Domain Techniques" lead to biased estimates (Wellstead 1981). This can easily be seen from the ETFE (15) applied to the feedback setup in Fig. 2

$$\hat{G}(\Omega_k) = \frac{G(\Omega_k)G_{\text{act}}(\Omega_k)R(k) + V(k)}{G_{\text{act}}(\Omega_k)R(k) - G_{\text{fb}}(\Omega_k)V(k)} \quad (26)$$

where $G_{\text{act}}(\Omega_k)$ and $G_{\text{fb}}(\Omega_k)$ are, respectively, the actuator and feedback dynamics. From (26), it follows that in those frequency bands where the process noise $V(k)$ dominates, one rather estimates minus the inverse of the feedback dynamics instead of the plant FRF. On the other hand, at those frequencies where the reference signal injects most power, the ETFE (26) will be close to the plant FRF.
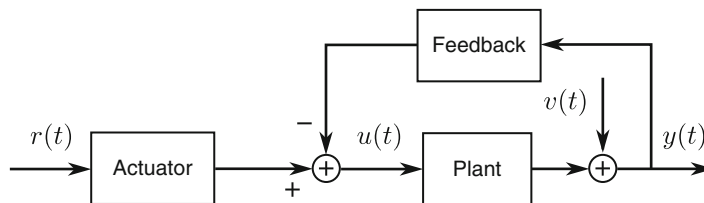
Nonparametric Techniques in System Identification, Fig. 2 Plant operating in closed loop: $r(t)$ is the external reference signal, the known input $u(t)$ depends on the process noise $v(t)$, and $y(t)$ is the noisy output observation
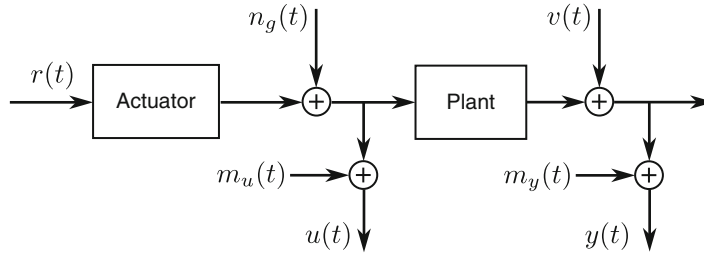
Nonparametric Techniques in System Identification, Fig. 3 Errors-in-variables framework: $r(t)$ is the external reference signal; $n_g(t)$ is the generator noise; $m_u(t)$, $m_y(t)$ are the input and output measurement errors; $v(t)$ is the process noise; and $u(t)$, $y(t)$ are the noisy input, noisy output observations

If a known external reference signal is available, then the bias is avoided via the indirect method proposed in Wellstead (1981)

$$G(\Omega) = \frac{S_{yr}(\Omega)/S_{rr}(\Omega)}{S_{ur}(\Omega)/S_{rr}(\Omega)} = \frac{S_{yr}(\Omega)}{S_{ur}(\Omega)}. \quad (27)$$

The basic idea consists in modeling the feedback setup (see Fig. 2) from the known reference to the input and output simultaneously. This reduces the single-input, single-output closed loop problem to a single-input, two-output open loop problem. Since the process noise $v(t)$ is independent of the reference signal $r(t)$, the direct estimate of the single-input, two-output FRF is unbiased. Calculating the ratio of the two FRFs finally gives the indirect estimate (27). This procedure can be applied to any of the direct methods of sections "Nonparametric Time-Domain Techniques" and "Nonparametric Frequency-Domain Techniques." Proceeding in this way, unstable plants operating in a stabilizing feedback loop can also be handled.

If the excitation is *periodic*, then the process noise $v(t)$ is independent of the periodic part of the input $u(t)$, and the ETFE (15) converges to the true value as the number of periods P tends to infinity (Pintelon and Schoukens 2012, Section 2.5). Hence, in the periodic case, no external reference is needed.

Noisy Input, Noisy Output Observations

The key difficulty of estimating the FRF of a plant excited by a *nonperiodic signal* from noisy input, noisy output observations (see Fig. 3) is that the input autopower spectrum in (17) is biased. Indeed, due to the noise on the input, $S_{uu}(\Omega)$ is too large, resulting in too small direct FRF estimates. This is true for all direct FRF approaches in sections "Nonparametric Time-Domain Techniques" and "Nonparametric Frequency-Domain Techniques." Applying the indirect method of section "Systems Operating in Feedback" removes the bias because the noise on the input is independent of the reference signal (e.g., see (27)). Proceeding in this way, the closed loop case (see Fig. 2) with noisy input, noisy output observations is also solved by the indirect method.

If the excitation is *periodic*, then the mean value of the input-output DFT spectra over the P consecutive periods converges to the their true values as P tends to infinity (Pintelon and Schoukens 2012, Section 2.5). Hence, the ETFE (15) is still consistent, and no external reference is needed. The same conclusion is valid for systems operating in feedback.

Nonlinear Systems

The classes of nonlinear systems considered are those systems whose steady-state response to a periodic input is periodic with the same period as the input. It excludes phenomena such as chaos and subharmonics but allows for hard nonlinearities such as saturation, dead zones, and clipping.

The classes of excitations considered are stationary random signals with a specified power spectrum and probability density function. An important special case is the class of
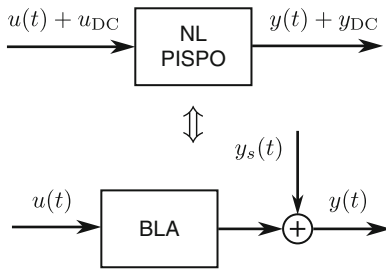
Nonparametric Techniques in System Identification, Fig. 4 Best linear approximation (*BLA*) of a nonlinear (*NL*) period in, same period out (*PISPO*) system, excited by a zero-mean random signal $u(t)$ with a given power spectrum and probability density function. $y(t)$ is the zero-mean part of the actual output of the nonlinear system. u_{DC} and y_{DC} are the DC levels of the actual input and output of the nonlinear system. The zero-mean output residual $y_s(t)$ is uncorrelated with – but not independent of – the input $u(t)$

Gaussian excitation signals with a specified power spectrum. This class includes random phase multisines (a sum of harmonically related sinewaves with user-specified amplitudes and random phases) with the same Riemann equivalent power spectrum (Pintelon and Schoukens 2012, Section 4.2).

Consider a nonlinear (NL) period in, same period out (PISPO) system excited by a random excitation belonging to a particular class (see Fig. 4). The FRF (17), where the expected value is taken w.r.t. the random realization of the excitation, is the best (in mean square sense) linear approximation (BLA) of the nonlinear PISPO system, because the difference $y_s(t)$ between the actual output of the nonlinear system (DC value excluded) and the output predicted by the linear approximation is uncorrelated with the input $u(t)$ (Enqvist and Ljung 2005). Although uncorrelated with the input, the output residual $y_s(t)$ still depends on $u(t)$. If the NL PISPO system operates in feedback (see Fig. 2), then the indirect method (27) is used for calculating the BLA, and the output residual $y_s(t)$ is uncorrelated with – but not independent of – the reference signal $r(t)$ (Fig. 2).

For the class of Gaussian excitation signals, it can be shown that the DFT spectrum $Y_S(k)$ of $y_s(t)$ has the following properties (Pintelon and Schoukens 2012, Section 3.4.4):

1. $Y_S(k)$ has zero-mean value: $\mathbb{E}\{Ys(k)\} = 0$.
2. $Y_S(k)$ it is uncorrelated with – but not independent of $-U(k) : \mathbb{E}\{Y(k)\overline{U(k)}\} = 0$.
3. $Y_S(k)$ is asymptotically ($N \to \infty$) normally distributed.
4. $Y_S(k)$ is asymptotically ($N \to \infty$) uncorrelated over the frequency.

These second-order properties are exactly the same as those of a filtered white noise disturbance, except that the noise is independent of the input. It shows that it is impossible to distinguish the nonlinear distortions $y_s(t)$ from the disturbing noise $v(t)$ in FRF measurements using stationary random excitations (only second-order statistics are involved in (22)–(24)).

Using random phase multisines, it is possible to detect and quantify the nonlinear distortions because $y_s(t)$ is then periodically related to the input $u(t)$ (property of the NL PISPO system). Indeed, analyzing the FRF over consecutive signal periods quantifies the noise variance $v(t)$ ($y_s(t)$ does not change over the periods), while analyzing the FRF over different random phase realizations of the input quantifies the sum of the noise variance and the variance of the nonlinear distortions ($y_s(t)$ depends on the random phase realization of the input). Subtracting both variances gives an estimate of the variance of the nonlinear distortions. While this variance quantifies exactly the variability of the nonparametric FRF estimate due to the nonlinear distortions, it can (significantly) underestimate the variability of a parametric plant model. The basic reason for this is that the true variance of the parametric plant model also depends on the nonzero higher (>2) order moments between the input $u(t)$ and the nonlinear distortions $y_s(t)$.

User Choices

There is no clear answer to the question which of the presented techniques is the best. It strongly depends on the intended use of the nonparametric estimates and the particular application handled. For example, the intended use can be:

1. A smooth representation of the FRF
2. Use of the nonparametric estimates as an intermediate step for parametric modeling of the plant

In the first case, one should opt for the minimum mean square error solution, while in the second case, it is crucial that the nonparametric estimates are unbiased, possibly at the price of an increased variance. Indeed, the parametric plant modeling step cannot eliminate the bias error in the nonparametric estimates while it suppresses the variance error.

The application-dependent answers to the following questions strongly influence the choice and the settings of the method used:

1. Is a large frequency resolution needed and/or is leakage the dominant error?
2. Is the noise or the leakage error dominant?
3. Is it necessary to detect and quantify the nonlinear behavior?

If the answer to the first question is yes, then one should opt for one of the advanced methods (section "Advanced Methods") or use the spectral analysis estimates (section "Spectral Analysis Method") with a small number M of subrecords. On the other hand, if the noise error is dominant, then M in (22)–(24) should be chosen as large as possible. To detect and quantify the nonlinear effects, one should use periodic signals (random phase multisines) combined with the ETFE (section "Empirical Transfer Function Estimation").

Finally, comparing the different nonparametric techniques is also not straightforward because of their different

1. Frequency resolution
2. Quality of the estimated noise model
3. Correlation length over the frequency

The latter is set by the spectral width of the window used in the spectral analysis method and the local bandwidth in the advanced methods.

Guidelines

While the previous sections give well-established facts about the different nonparametric techniques,

in this section, we provide some advices/guidelines based on our personal interpretation of these facts:

- Always store the reference signal together with the observed input-output signals. The knowledge of the reference signal allows one to solve nonparametrically the closed loop and errors-in-variables problems.
- Whenever possible use periodic excitation signals (random phase multisines): they allow one to estimate from one experiment the FRF, the noise level, and the level of the nonlinear distortions. As such the deviation of the true dynamic behavior from the ideal linear time-invariant framework is quantified.
- Select one of the advanced methods if frequency resolution is of prime interest.
- If the goal of the identification experiment is to minimize the prediction error, then the Gaussian process regression method is a very promising approach.
- For lowly damped systems and a limited frequency resolution, the local rational method is a good candidate solution.
- Use a minimum mean square solution for a smooth representation of the FRF.
- Choose unbiased nonparametric estimates for use in parametric plant modeling (estimation, validation, and model selection).
- When comparing nonparametric techniques, always take into account all aspects of the estimates: the bias and variance of the FRF and noise model, the frequency resolution, and the correlation length over the frequency.

Summary and Future Directions

Nonparametric techniques are very useful because they simplify the parametric plant modeling in the initial selection of the model complexity and in the detection of unmodeled dynamics. The classical correlation and spectral analysis methods developed in the 1950s and refined till the 1980s are still widely used. Recently, advanced time- and frequency-domain

methods have been developed which all try to minimize the sensitivity (bias and variance) of the nonparametric estimates to disturbing noise, nonlinear distortion, and transient (leakage) errors.

The renewed research interest in nonparametric techniques should be continued to handle the following challenging problems: short data sets, missing data, detection and quantification of time-variant behavior, modeling of time-variant dynamics, and modeling of nonlinear dynamics.

Cross-References

▶ Frequency Domain System Identification
▶ Frequency-Response and Frequency-Domain Models
▶ System Identification: An Overview
▶ System Identification Techniques: Convexification, Regularization, and Relaxation

Recommended Reading

The classical correlation (see section "Correlation Methods") and spectral analysis (see section "Spectral Analysis Method") methods are well covered by the text books listed below. The recommended reading list includes the basic papers on the spectral analysis methods (Blackman and Tukey 1958; Welch 1967; Wellstead 1981) and the most recent developments described in sections "Gaussian Process Regression" and "Advanced Methods."

Bibliography

Antoni J, Schoukens J (2007) A comprehensive study of the bias and variance of frequency-response-function measurements: optimal window selection and overlapping strategies. Automatica 43(10):1723–1736

Bendat JS, Piersol AG (1980) Engineering applications of correlations and spectral analysis. Wiley, New York

Blackman RB, Tukey JW (1958) The measurement of power spectra from the point of view of communications engineering – Part II. Bell Syst Tech J 37(2):485–569

Brillinger DR (1981) Time series: data analysis and theory. McGraw-Hill, New York

Broersen PMT (2004) Mean square error of the empirical transfer function estimator for stochastic input signals. Automatica 40(1):95–100

Carter GC, Nuttall AH (1980) On the weighted overlapped segment averaging method for power spectral estimation. Proc IEEE 68(10):1352–1353

Chen T, Ohlsson H, Ljung L (2012) On the estimation of transfer functions, regularizations and Gaussian processes – revisited. Automatica 48(8): 1525–1535

Enqvist M, Ljung L (2005) Linear approximations of nonlinear FIR systems for separable input processes. Automatica 41(3):459–473

Eykhoff P (1974) System identification. Wiley, New York

Godfrey K (1993) Perturbation signals for system identification. Prentice-Hall, Englewood Cliffs

Hägg P, Hjalmarsson H (2012) Non-parametric frequency response function estimation using transient impulse response modelling. Paper presented at the 16th IFAC symposium on system identification, Brussels, 11–13 July, pp 43–48

Heath WP (2007) Choice of weighting for averaged nonparametric transfer function estimates. IEEE Trans Autom Control 52(10):1914–1920

Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River

McKelvey T, Guérin G (2012) Non-parametric frequency response estimation using a local rational model. Paper presented at the 16th IFAC symposium on system identification, Brussels, 11–13 July, pp 49–54

Pillonetto G, Chiuso A, De Nicolao G (2011) Prediction error identification of linear systems: a nonparametric Gaussian regression approach. Automatica 47(2):291–305

Pintelon R, Schoukens J (2012) System identification: a frequency domain approach, 2nd edn. IEEE, Piscataway/Wiley, Hoboken

Schoukens J, Rolain Y, Pintelon R (2006) Analysis of windowing/leakage effects in frequency response function measurements. Automatica 42(1):27–38

Stenman A, Gustafsson F, Rivera DE, Ljung L, McKelvey T (2000) On adaptive smoothing of empirical transfer function estimates. Control Eng Pract 8(11):1309–1315

Welch PD (1967) The use of the fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Trans Audio Electroacoust 15(2): 70–73

Wellstead PE (1981) Non-parametric methods of system identification. Automatica 17(1):55–69

N

Numerical Methods for Continuous-Time Stochastic Control Problems

George Yin
Department of Mathematics, Wayne State University, Detroit, MI, USA

Abstract

This expository article provides a brief review of numerical methods for stochastic control in continuous time. It concentrates on the methods of Markov chain approximation for controlled diffusions. Leaving most of the technical details out with the broad general audience in mind, it aims to serve as an introductory reference or a user's guide for researchers, practitioners, and students who wish to know something about numerical methods for stochastic control.

Keywords

Markov chain approximation; Numerical methods; Stochastic control

Introduction

This expository article provides a brief review of numerical methods for stochastic control in continuous time. Leaving most of the technical details out with the broad general audience in mind, it aims to serve as an introductory reference for researchers, practitioners, and students, who wish to know something about numerical methods for stochastic controls.

The study of stochastic control has witnessed tremendous progress in the last few decades; see, for example, Fleming and Rishel (1975), Fleming and Soner (1992), Kushner (1977), and Yong and Zhou (1999) among others, for fundamentals of stochastic controls as well as historical remarks. Much of the development has been accompanied by the needs and progress in science, engineering, as well as finance. Typically, the problems are highly nonlinear, so a closed-form solution is very difficult to obtain. As a result, designing feasible numerical algorithms becomes vitally important. Among the many approximation methods, the Markov chain approximation methods have shown most promising features. Primarily for treating diffusions, the Markov chain approximation method was initiated in the 1970s (Kushner 1977) and substantially developed further in Kushner (1990b) and Kushner and Dupuis (1992). Nowadays, such method are used for more complex jump diffusions, or systems with random switchings. There were also efforts to incorporate the methods into an expert system so that the methods can be placed into an easily usable tool box (Chancelier et al. 1986, 1987). In addition to the existing applications in a wide variety of engineering problems, recently applications include such areas as insurance, quantile hedging for guaranteed minimum death benefits, dividend payment and investment strategies with capital injection, singular control, risk management, portfolio selection with bounded constraints, and production planning and manufacturing problems; see Jin et al. (2011, 2012, 2013), Sethi and Zhang (1994), and Yin et al. (2009) and references therein.

Let us begin with the controlled diffusion problem. We wish to minimize the cost function defined by

$$J(x, u(\cdot)) = E_x\left[\int_0^\tau R(X(t), u(t))dt + B(X(\tau))\right],$$
(1)

with the $\mathbb{R}^r$-valued process $X(t)$ defined by the solution of the stochastic differential equation

$$dX(t) = b(X(t), u(t))dt + \sigma(X(t))dW,$$
$$X(0) = x$$
(2)

where $x \in \mathbb{R}^r$, $u(\cdot)$ is a U-valued, measurable process with $U \subset \mathbb{R}^d$ being a compact control set, $W(\cdot)$ is an r-dimensional standard Brownian motion, and τ is the first exit time of the diffusion from a bounded domain D, that is, $\tau = \min\{t : X(t) \notin D^0\}$ with D^0 denoting the interior of D, $b(\cdot, \cdot) : \mathbb{R}^r \times \mathbb{R}^d \mapsto \mathbb{R}^r$, $\sigma(\cdot) : \mathbb{R}^r \mapsto \mathbb{R}^r \times \mathbb{R}^r$, and $R(\cdot, \cdot) : \mathbb{R}^r \times \mathbb{R}^d \mapsto \mathbb{R}$ and $B(\cdot) : \mathbb{R}^r \mapsto \mathbb{R}$. In the above, $b(\cdot)$ is the control-dependent drift, $\sigma(\cdot)$ is the diffusion matrix, $R(\cdot)$ is the running cost, and $B(\cdot)$ is the terminal or boundary cost. Throughout the entry, we assume that the stopping time $\tau < \infty$ with probability one (w.p.1) for simplicity. Denote the value function by $V(x) = \inf_u J(x, u(\cdot))$, where the inf is taken over all admissible controls. Write the transpose of $Y \in \mathbb{R}^{d_1 \times d_2}$ as Y' with $d_1, d_2 \geq 1$, $a(x) = \sigma(x)\sigma'(x)$, and define the generator of the controlled Markov process by

$$\mathcal{L}^u f(x) = \frac{1}{2}\mathrm{tr}(a(x)f_{xx}(x)) + b'(x, u)f_x(x), \tag{3}$$

for a suitably smooth function $f(\cdot)$, where $f_x(\cdot)$ and $f_{xx}(\cdot)$ denote the gradient and Hessian of $f(\cdot)$, respectively. Note that the operator is control dependent. Using ∂D to denote the boundary of D, then the associated Hamilton-Jacobi-Bellman (HJB) equation satisfied by the value function is given by

$$\begin{cases} \inf_u[\mathcal{L}^u V(x) + R(x, u)] = 0, \; x \in D^0, \\ V(x) = B(x), \; x \in \partial D. \end{cases} \tag{4}$$

The subject matter of this article is to solve the optimal stochastic control problem numerically.

The rest of the entry is arranged as follows. Section "Markov Chain Approximation" focuses on Markov chain approximation. It illustrates how one can construct the controlled Markov chain in discrete time for the approximation of the continuous-time stochastic control problems. Section "Illustration: A One-Dimensional Problem" uses a one-dimensional case as an example

for illustration. Section "Numerical Computation" discusses the implementation issues. We conclude the entry with a few further remarks.

Markov Chain Approximation

The main idea was initiated in Kushner (1977) and streamlined, extended, and further developed in Kushner and Dupuis (1992). An earlier paper describing how to discretize the elliptic HJB equation and then interpret it according to a controlled Markov chain can be found in Kushner and Kleinman (1968). This section illustrates the Markov chain approximation methods with simple setup. The reader is suggested to read the references mentioned above for a comprehensive treatment. To begin, let $h > 0$ be a small "step size" in the approximation. Instead of the domain D, we need to work with a finite set to ensure computational feasibility. Set $\mathbb{R}_h^r$ to be r-dimensional lattice cube, i.e., $\mathbb{R}_h^r = \{\dots, -2h, -h, 0, h, 2h, \dots\}^r$ (an r-dimensional product of the indicated set). Denote the interior of D by D^0, and define $D_h^0 = D^0 \cap \mathbb{R}_h^r$. We shall construct a controlled, discrete-time Markov chain, whose transition probabilities have desired properties in line with the controlled diffusion and whose values are in D_0^h. Suppose that $\{\alpha_n^h\}$ is a time-homogeneous, discrete-time, controlled Markov chain with finite state space D_0^h and transition probabilities $P = (p(x, y|v))$ with $x, y \in D_0^h$. Here we only consider the case that the Markov chain has a finite state space. This is sufficient for our computational purposes. At any time n, the control action is a random variable denoted by u_n^h taking values in a compact set U. Set the interpolation interval by $\Delta t^h(x, v) > 0$ and write $\Delta t_n^h = \Delta t^h(\alpha_n, u_n^h)$ such that $\sup_{x,v} \Delta t^h(x, v) \to 0$ as $h \to 0$ but $\inf_{x,v} \Delta t^h(x, v) > 0$ for each $h > 0$. The control is admissible if the Markov property $P(\alpha_{n+1}^h = y|\alpha_i^h, u_i^h; i \leq n) = P(\alpha_{n+1}^h = y|\alpha_n^h, u_n^h) = P(\alpha_n^h, y|u_n^h)$ holds. Use U^h to denote the collection of controls, which

N

are determined by a sequence of measurable functions $F_n^h(\cdot)$ such that $u_n^h = F_n^h(\alpha_k^h, k \leq n; u_k^h, k < n)$. Denote the conditional expectation given $\{\alpha_j^h, u_j^h : j \leq n, \alpha_n^h = x, u_n^h = v\}$ by E_n^h. We say that a control policy is locally consistent if

$$E_n^h \alpha_n^h = b(x, v) \Delta t^h(x, v) + o(\Delta^h(x, v)),$$

$$E_n^h [\Delta \alpha_n^h - E_n^h \Delta \alpha_n^h][\Delta \alpha_n^h - E_n^h \Delta \alpha_n^h]' = a(x) \Delta t^h(x, v) + o(\Delta t^h(x, v)), \tag{5}$$

$$a(x) = \sigma(x)\sigma'(x), \quad |\Delta \alpha_n^h| \to 0 \text{ as } h \to 0 \text{ uniformly in } n, \omega,$$

where $\Delta \alpha_n^h = \alpha_{n+1}^h - \alpha_n^h$. The meaning of the local consistency can be seen from the corresponding controlled diffusion (2) (with $X(0) = x$ and $u(t) = v$ for $t \in [0, \delta]$, where $\delta > 0$ is a small parameter) in that $E_x(X(\delta) - x) = b(x, v)\delta + o(\delta)$, $E_x[X(\delta) - x][X(\delta) - x]' = a(x)\delta + o(\delta)$. Let ν_h be the first time that $\{\alpha_n^h\}$ leaves the set D_h^0. We have an approximation for the cost function of the controlled diffusion (1) given by

$$J^h(x, u^h) = E_x^{u^h} \left[\sum_{j=0}^{\nu_h - 1} R(\alpha_j^h, u_j^h) \Delta t_j^h + B(\alpha_{\nu_h}^h) \right]. \tag{6}$$

Define $t_n^h = \sum_{j=0}^{n-1} \Delta t_j^h$ and the continuous-time interpolations $\alpha^h(t) = \alpha_n^h$, $u_n^h = u_n^h$ for $t \in [t_n^h, t_{n+1}^h)$. Define the first exit time of $\alpha^h(\cdot)$ from D_h^0 by $\tau_h = t_{\nu_h}^h$. Corresponding to the continuous-time problems, the first term on the right-hand side of (6) represents the running cost and the last term gives the terminal cost. Denote the value function by $V^h(x)$. Then it satisfies the dynamic programming equation

$$V^h(x) = \begin{cases} \inf_{v \in U^h} [R(x, v) \Delta t^h(x, v) \\ \qquad + \sum_y p^h(x, y|v) V^h(y)], \ x \in D_h^0, \\ B(x), \ x \notin D_h^0. \end{cases} \tag{7}$$

Proving the convergence of the numerical algorithms is an important task. This requires the use of local consistency, interpolation of the approximating sequences in continuous time, as well as martingale representation. The proof is facilitated by the use of the so-called relaxed controls (Kushner and Dupuis 1992, p. 267), which enables us to characterize the limit under the framework of weak convergence. The detailed argument is beyond the scope of this entry. We refer the reader to Kushner and Dupuis (1992, Chapter 10) for further reading on the proof of convergence and the conditions needed.

Illustration: A One-Dimensional Problem

In this section, we use a one-dimensional example to illustrate the Markov chain approximation methods, which enables us to present the results with a better visualization. Consider (2) with $x \in \mathbb{R}$. We proceed to find the transition probabilities and interpolation intervals for the Markov chain $\{\alpha_n^h\}$. To construct a controlled Markov chain that is locally consistent, we first consider a special case, namely, the control space has only one admissible control $u^h \in U^h$. In this case, min in (7) can be removed. Discretize the HJB equation using upwind finite difference method with step size $h > 0$ by

$$V(x) \to V^h(x)$$

$$V_x(x) \to \frac{V^h(x + h) - V^h(x)}{h} \text{ for } b(x, v) > 0,$$

$$V_x(x) \to \frac{V^h(x) - V^h(x - h)}{h} \text{ for } b(x, v) < 0,$$

$$V_{xx}(x) \to \frac{V^h(x + h) - 2V^h(x) + V^h(x - h)}{h^2}.$$

For $x \in D_h^0$, it leads to

$$\frac{V^h(x+h) - V^h(x)}{h} b^+(x, v) - \frac{V^h(x) - V^h(x-h)}{h} b^-(x, v)$$
$$+ \frac{a(x)}{2} \frac{V^h(x+h) - 2V^h(x) + V^h(x-h)}{h^2} + R(x, v) = 0,$$

where b^+ and b^- are the positive and negative parts of b, respectively. Comparing with the dynamic programming equation, we obtain the transition probabilities

$$p^h(x, x+h)|v) = \frac{(a(x)/2) + hb^+(x, v)}{\tilde{\Delta}},$$
$$p^h(x, x-h)|v) = \frac{(a(x)/2) + hb^-(x, v)}{\tilde{\Delta}},$$
$$p^h(\cdot) = 0, \quad \text{otherwise}, \quad \Delta t^h(x, v) = \frac{h^2}{\tilde{\Delta}},$$

with $\tilde{\Delta} = a(x) + h|b(x, v)|$ being well defined. With the transition probabilities given above, we can proceed to verify the local consistency by straight forward calculations and prove the desired convergence.

Numerical Computation

To numerically approximate the controlled diffusions, frequently used methods are either value iterations or policy iterations (iteration in policy space). Using Markov chain approximation in conjunction with either value iteration or iteration in policy space, we can further obtain a sequence of value functions $\{V^{h,n}\}$ such that $V^{h,n} \to V^h$ as $n \to \infty$. The procedures can be described as follows.

Value Iteration
1. Given a tolerance $\varepsilon > 0$, set $n = 0$; for $x \in D_h^0$, set $V^{h,0} = $ constant (for instance, 0).
2. Using $V^{h,n}$ obtained in (7) to obtain $V^{h,n+1}$.
3. If $|V^{h,n+1} - V^{h,n}| > \varepsilon$, go to Step 3 above with $n \to n + 1$.

Policy Iteration
1. Given a tolerance $\varepsilon > 0$, set $n = 0$; for $x \in D_h^0$, take an initial control $u_0^h(x) = $ constant. Use $u_0^h(x)$ in lieu of v, solve (7) to find $V^{h,0}(\cdot)$.

2. Find an improved control by
$u^{h,n+1}(x) := \text{argmin}_{v \in U^h}[\sum_y p^h((x, y)|v) V^{h,n}(y) + R(x, v)\Delta t^h(x, v)].$
3. Find $V^{h,n+1}(\cdot)$ with $u^{h,n+1}(\cdot)$ by solving (7). If $|V^{h,n+1} - V^{h,n}| > \varepsilon$, go to Step 2 above with $n \to n + 1$.

Further Remarks

Variations of the Problems. Variants of the problems can be considered. For example, one may consider nonlinear filtering problems or singularly perturbed control and filtering problems. For problems arising in manufacturing systems, one often needs to treat controlled Markov chain with no diffusion terms. Such a case can also be handled by the Markov chain approximation methods; see Sethi and Zhang (1994) for the problem and Yin and Zhang (2013, Chapter 9) for the numerical methods. In this article, we mainly discussed the approach by using probabilistic approach for getting the weak convergence of the interpolations of the controlled Markov chain. One can also use the so-called viscosity solution methods to treat the convergence; see Barles and Souganidis (1991) (also Kushner and Dupuis 1992, Chapter 11).

Variance Control. In this entry, only drift involves control term. When the diffusion term is also subject to controls, the problem becomes more difficult. In Peng (1990), the idea of using backward stochastic differential equations was initiated, which had significant impact in the development of such stochastic control problems. Detailed discussions can be found in Yong and Zhou (1999). The numerical problems for diffusion term involving controls can also be treated; see Kushner (2000) for further discussion. In this case, the so-called numerical noise or numerical viscosity can be introduced, so care must be taken.

N

Complex Models Involving Jump and Switching. Note that only controlled diffusions are considered in this entry. More complex models such as controlled jump diffusions (Kushner and Dupuis 1992), switching diffusions (Yin and Zhu 2010), and switching jump diffusions can be treated (Song et al. 2006). Differential games can also be treated (Kushner 2002; Song et al. 2008).

Differential Delay Systems. Stochastic differential delay systems may come into play. The corresponding numerical algorithms have been studied extensively in Kushner (2008). Due to their inherent infinite dimensionality, a main issue here concerns suitable finite approximation to the memory segments.

Rates of Convergence. This entry mainly discusses the convergence of the approximation methods. There is also much interest in ascertaining rates of convergence. Such effort goes back to the paper Menaldi (1989) (see also Zhang 2006). Subsequently, it has been resurgent effort in dealing with this issue from a nonlinear partial differential equation point of view; see Krylov (2000). Our recent work Song and Yin (2009) complements the study by providing a probabilistic approach for treating switching diffusions.

Stochastic Approximation. In certain optimal control problems, the optimal controls or near-optimal controls turn out to be of threshold type. An alternative way of solving such problems leading to at least suboptimal or near-optimal control is to use a stochastic approximation approach; see Kushner and Yin (2003) for a comprehensive treatment of stochastic approximation algorithms. Some successful examples include manufacturing systems (Yin and Zhang 2013, Section 9.3) and liquidation decision making (Yin et al. 2002).

Cross-References

▶ Stochastic Dynamic Programming
▶ Stochastic Maximum Principle

Bibliography

Barles G, Souganidis P (1991) Convergence of approximation schemes for fully nonlinear second order equations. J Asymptot Anal 4:271–283

Bertsekas DP, Castanon DA (1989) Adaptive aggregation methods for infinite horizon dynamic programming. IEEE Trans Autom Control 34:589–598

Chancelier P, Gomez C, Quadrat J-P, Sulem A, Blankenship GL, La Vigna A, MaCenary DC, Yan I (1986) An expert system for control and signal processing with automatic FORTRAN program generation. In: Mathematical systems symposium, Stockholm. Royal Institute of Technology, Stockholm

Chancelier P, Gomez C, Quadrat J-P, Sulem A (1987) Automatic study in stochastic control. In: Fleming W, Lions PL (eds) IMA volume in mathematics and its applications, vol 10. Springer, Berlin

Crandall MG, Lions PL (1983) Viscosity solutions of Hamilton-Jacobi equations. Trans Am Math Soc 277:1–42

Crandall MG, Ishii H, Lions PL (1992) User's guide to viscosity solutions of second order partial differential equations. Bull Am Math Soc 27:1–67

Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York

Fleming WH, Soner HM (1992) Controlled Markov processes and viscosity solutions. Springer, New York

Jin Z, Wang Y, Yin G (2011) Numerical solutions of quantile hedging for guaranteed minimum death benefits under a regime-switching-jump-diffusion formulation. J Comput Appl Math 235:2842–2860

Jin Z, Yin G, Zhu C (2012) Numerical solutions of optimal risk control and dividend optimization policies under a generalized singular control formulation. Automatica 48:1489–1501

Jin Z, Yang HL, Yin G (2013) Numerical methods for optimal dividend payment and investment strategies of regime-switching jump diffusion models with capital injections. Automatica 49:2317–2329

Krylov VN (2000) On the rate of convergence of finite-difference approximations for Bellman's equations with variable coefficients. Probab Theory Relat Fields 117:1–16

Kushner HJ (1977) Probability methods for approximation in stochastic control and for elliptic equations. Academic, New York

Kushner HJ (1990a) Weak convergence methods and singularly perturbed stochastic control and filtering problems. Birkhäuser, Boston

Kushner HJ (1990b) Numiercal methods for stochastic control problems in continuous time. SIAM J Control Optim 28:999–1048

Kushner HJ (2000) Consistency issues for numerical methods for variance control with applications to optimization in finance. IEEE Trans Autom Control 44:2283–2296

Kushner HJ (2002) Numerical approximations for stochastic differential games. SIAM J Control Optim 40:457–486

Kushner HJ (2008) Numerical methods for controlled stochastic delay systems. Birkhäuser, Boston

Kushner HJ, Dupuis PG (1992) Numerical methods for stochastic control problems in continuous time. Springer, New York

Kushner HJ, Kleinman AJ (1968) Numerical methods for the solution of the degenerate nonlinear elliptic equations arising in optimal stochastic control theory. IEEE Trans Autom Control AC-13: 344–353

Kushner HJ, Yin G (2003) Stochastic approximation and recursive algorithms and applications, 2nd edn. Springer, New York

Menaldi J (1989) Some estimates for finite difference approximations. SIAM J Control Optim 27:579–607

Peng S (1990) A general stochastic maximum principle for optimal control problems. SIAM J Control Optim 28:966–979

Sethi SP, Zhang Q (1994) Hierarchical decision making in stochastic manufacturing systems. Birkhäuser, Boston

Song QS, Yin G (2009) Rates of convergence of numerical methods for controlled regime-switching diffusions with stopping times in the costs. SIAM J Control Optim 48:1831–1857

Song QS, Yin G, Zhang Z (2006) Numerical method for controlled regime-switching diffusions and regime-switching jump diffusions. Automatica 42: 1147–1157

Song QS, Yin G, Zhang Z (2008) Numerical solutions for stochastic differential games with regime switching. IEEE Trans Autom Control 53:509–521

Warga J (1962) Relaxed variational problems. J Math Anal Appl 4:111–128

Yan HM, Yin G, Lou SXC (1994) Using stochastic optimization to determine threshold values for control of unreliable manufacturing systems. J Optim Theory Appl 83:511–539

Yin G, Zhang Q (2013) Continuous-time Markov chains and applications: a two-time-scale approach, 2nd edn. Springer, New York

Yin G, Zhu C (2010) Hybrid switching diffusions: properties and applications. Springer, New York

Yin G, Liu RH, Zhang Q (2002) Recursive algorithms for stock liquidation: a stochastic optimization approach. SIAM J Optim 13:240–263

Yin G, Jin H, Jin Z (2009) Numerical methods for portfolio selection with bounded constraints. J Comput Appl Math 233:564–581

Yong J, Zhou XY (1999) Stochastic controls: Hamiltonian systems and HJB equations. Springer, New York

Zhang J (2006) Rate of convergence of finite difference approximations for degenerate ODEs. Math Comput 75:1755–1778

Numerical Methods for Nonlinear Optimal Control Problems

Lars Grüne
Mathematical Institute, University of Bayreuth, Bayreuth, Germany

Abstract

In this article we describe the three most common approaches for numerically solving nonlinear optimal control problems governed by ordinary differential equations. For computing approximations to optimal value functions and optimal feedback laws, we present the Hamilton-Jacobi-Bellman approach. For computing approximately optimal open-loop control functions and trajectories for a single initial value, we outline the indirect approach based on Pontryagin's maximum principle and the approach via direct discretization.

Keywords

Direct discretization; Hamilton-Jacobi-Bellman equations; Optimal control; Ordinary differential equations; Pontryagin's maximum principle

Introduction

This article concerns optimal control problems governed by nonlinear ordinary differential equations of the form

$$\dot{x}(t) = f(x(t), u(t)) \tag{1}$$

with $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. We assume that for each initial value $x \in \mathbb{R}^n$ and measurable control function $u(\cdot) \in L^\infty(\mathbb{R}, \mathbb{R}^m)$ there exists a unique solution $x(t) = x(t, x, u(\cdot))$ of (1) satisfying $x(0, x, u(\cdot)) = x$.

Given a state constraint set $X \subseteq \mathbb{R}^n$ and a control constraint set $U \subseteq \mathbb{R}^m$, a running cost $g : X \times U \to \mathbb{R}$, a terminal cost $F : X \to U$, and

a discount rate $\delta \geq 0$, we consider the optimal control problem

$$\underset{u(\cdot) \in \mathcal{U}^T(x)}{\text{minimize}} \ J^T(x, u(\cdot)) \qquad (2)$$

where

$$J^T(x, u(\cdot)) := \int_0^T e^{-\delta s} g(x(s, x, u(\cdot)), u(s)) ds$$
$$+ \ e^{-\delta T} F(x(T, x, u(\cdot))) \qquad (3)$$

and

$$\mathcal{U}^T(x) :=$$
$$\left\{ u(\cdot) \in L^\infty(\mathbb{R}, U) \ \middle| \ \begin{matrix} x(s, x, u(\cdot)) \in X \\ \text{for all } s \in [0, T] \end{matrix} \right\} \qquad (4)$$

In addition to this finite horizon optimal control problem, we also consider the infinite horizon problem in which T is replaced by "∞," i.e.,

$$\underset{u(\cdot) \in \mathcal{U}^\infty(x)}{\text{minimize}} \ J^\infty(x, u(\cdot)) \qquad (5)$$

where

$$J^\infty(x, u(\cdot)) := \int_0^\infty e^{-\delta s} g(x(s, x, u(\cdot)), u(s)) ds \qquad (6)$$

and

$$\mathcal{U}^\infty(x) :=$$
$$\left\{ u(\cdot) \in L^\infty(\mathbb{R}, U) \ \middle| \ \begin{matrix} x(s, x, u(\cdot)) \in X \\ \text{for all } s \geq 0 \end{matrix} \right\}, \qquad (7)$$

respectively.

The term "solving" (2)–(4) or (5)–(7) can have various meanings. First, the optimal value functions

$$V^T(x) = \inf_{u(\cdot) \in \mathcal{U}^T(x)} J^T(x, u(\cdot))$$

or

$$V^\infty(x) = \inf_{u(\cdot) \in \mathcal{U}^\infty(x)} J^\infty(x, u(\cdot))$$

may be of interest. Second, and often more importantly, one would like to know the optimal

control policy. This can be expressed in *open-loop* form $u^\star : \mathbb{R} \to U$, in which the function $u^\star$ depends on the initial value x and on the initial time which we set to 0 here. Alternatively, the optimal control can be computed in state- and time-dependent *closed-loop* form, in which a feedback law $\mu^\star : \mathbb{R} \times X \to U$ is sought. Via $u^\star(t) = \mu^\star(t, x(t))$, this feedback law can then be used in order to generate the time-dependent optimal control function for all possible initial values. Since the feedback law is evaluated along the trajectory, it is able to react to perturbations and uncertainties which may make $x(t)$ deviate from the predicted path. Finally, knowing $u^\star$ or $\mu^\star$, one can reconstruct the corresponding optimal trajectory by solving

$$\dot{x}(t) = f(x(t), u^\star(t)) \quad \text{or}$$
$$\dot{x}(t) = f(x(t), \mu^\star(t, x(t))).$$

Hamilton-Jacobi-Bellman Approach

In this section we describe the numerical approach to solving optimal control problems via Hamilton-Jacobi-Bellman equations. We first describe how this approach can be used in order to compute approximations to the optimal value function V^T and V^∞, respectively, and afterwards how the optimal control can be synthesized using these approximations. In order to formulate this approach for finite horizon T, we interpret $V^T(x)$ as a function in T and x. We denote differentiation w.r.t. T and x with subscript T and x, i.e., $V_x^T(x) = dV^T(x)/dx$, $V_T^T(x) = dV^T(x)/dT$ etc.

We define the Hamiltonian of the optimal control problem as

$$H(x, p) := \max_{u \in U} \{ -g(x, u) - p \cdot f(x, u) \},$$

with $x, p \in \mathbb{R}^n$, f from (1), g from (3) or (6), and "$\cdot$" denoting the inner product in $\mathbb{R}^n$. Then, under appropriate regularity conditions on the problem data, the optimal value functions V^T and V^∞ satisfy the first order partial differential equations (PDEs)

$$V_T^T(x) + \delta V^T(x) + H(x, V_x^T(x)) = 0$$

and

$$\delta V^\infty(x) + H(x, V_x^\infty(x)) = 0$$

in the viscosity solution sense. In the case of V^T, the equation holds for all $T \geq 0$ with the boundary condition $V^0(x) = F(x)$.

The framework of viscosity solutions is needed because in general the optimal value functions will not be smooth; thus, a generalized solution concept for PDEs must be employed (see Bardi and Capuzzo Dolcetta 1997). Of course, appropriate boundary conditions are needed at the boundary of the state constraint set X.

Once the Hamilton-Jacobi-Bellman characterization is established, one can compute numerical approximations to V^T or V^∞ by solving these PDEs numerically. To this end, various numerical schemes have been suggested, including various types of finite element and finite difference schemes. Among those, semi-Lagrangian schemes Falcone (1997) or Falcone and Ferretti (2013) allow for a particularly elegant interpretation in terms of optimal control synthesis, which we explain for the infinite horizon case.

In the semi-Lagrangian approach, one takes advantage of the fact that by the chain rule for $p = V_x^\infty(x)$ and constant control functions u, the identity

$$\delta V^\infty(x) - p \cdot f(x, u) = \frac{d}{dt}\Big|_{t=0} - (1 - \delta t) V^\infty$$
$$(x(t, x, u))$$

holds. Hence, the left-hand side of this equality can be approximated by the difference quotient

$$\frac{V^\infty(x) - (1 - \delta h) V^\infty(x(h, x, u))}{h}$$

for small $h > 0$. Inserting this approximation into the Hamilton-Jacobi-Bellman equation, replacing $x(h, x, u)$ by a numerical approximation $\tilde{x}(h, x, u)$ (in the simplest case, the Euler method $\tilde{x}(h, x, u) = x + hf(x, u)$), multiplying

by h, and rearranging terms, one arrives at the equation

$$V_h^\infty(x) = \min_{u \in U}\{hg(x, u)$$
$$+ (1 - \delta h) V_h^\infty(\tilde{x}(h, x, u))\}$$

defining an approximation $V_h^\infty \approx V^\infty$. This is now a purely algebraic dynamic programming-type equation which can be solved numerically, e.g., by using a finite element approach. The equation is typically solved iteratively using a suitable minimization routine for computing the "min" in each iteration (in the simplest case, U is discretized with finitely many values and the minimum is determined by direct comparison). We denote the resulting approximation of V^∞ by $\tilde{V}_h^\infty$. Here, approximation is usually understood in the L^∞ sense (see Falcone 1997 or Falcone and Ferretti 2013).

The semi-Lagrangian scheme is appealing for synthesis of an approximately optimal feedback because V_h^∞ is the optimal value function of the auxiliary discrete-time problem defined by $\tilde{x}$. This implies that the expression

$$\mu_h^\star(x) := \operatorname*{argmin}_{u \in U}\{hg(x, u)$$
$$+ (1 - \delta h) V_h^\infty(\tilde{x}(h, x, u))\},$$

is an optimal feedback control value for this discrete-time problem for the next time step, i.e., on the time interval $[t, t + h]$ if $x = x(t)$. This feedback law will be approximately optimal for the continuous-time control system when applied as a discrete-time feedback law, and this approximate optimality remains true if we replace V_h^∞ in the definition of $\mu_h^\star$ by its numerically computable approximation $\tilde{V}_h^\infty$. A similar construction can be made based on any other numerical approximation $\tilde{V}^\infty \approx V^\infty$, but the explicit correspondence of the semi-Lagrangian scheme to a discrete-time auxiliary system facilitates the interpretation and error analysis of the resulting control law.

The main advantage of the Hamilton-Jacobi approach is that it directly computes an approximately optimal feedback law. Its main

disadvantage is that the number of grid nodes needed for maintaining a given accuracy in a finite element approach to compute $\tilde{V}_h^\infty$ in general grows exponentially with the state dimension n. This fact – known as the *curse of dimensionality* – restricts this method to low-dimensional state spaces. Unless special structure is available which can be exploited, as, e.g., in the max-plus approach (see McEneaney 2006), it is currently almost impossible to go beyond state dimensions of about $n = 10$, typically less for strongly nonlinear problems.

Maximum Principle Approach

In contrast to the Hamilton-Jacobi-Bellman approach, the approach via Pontryagin's maximum principle does not compute a feedback law. Instead, it yields an approximately open-loop optimal control $u^\star$ together with an approximation to the optimal trajectory $x^\star$ for a fixed initial value. We explain the approach for the finite horizon problem. For simplicity of presentation, we omit state constraints in our presentation, i.e., we set $X = \mathbb{R}^n$ and refer to, e.g., Vinter (2000), Bryson and Ho (1975), or Grass et al. (2008) for more general formulations as well as for rigorous versions of the following statements.

In order to state the maximum principle (which, since we are considering a minimization problem here, could also be called minimum principle), we define the non-minimized Hamiltonian as

$$\mathcal{H}(x, p, u) = g(x, u) + p \cdot f(x, u).$$

Then, under appropriate regularity assumptions, there exists an absolutely continuous function $p : [0, T] \to \mathbb{R}^n$ such that the optimal trajectory $x^\star$ and the corresponding optimal control function $u^\star$ for (2)–(4) satisfy

$$\dot{p}(t) = \delta p(t) - \mathcal{H}_x(x^\star(t), p(t), u^\star(t)) \quad (8)$$

with terminal or transversality condition

$$p(T) = F_x(x^\star(T)) \quad (9)$$

and

$$u^\star(t) = \underset{u \in U}{\mathrm{argmin}}\, \mathcal{H}(x^\star(t), p(t), u), \quad (10)$$

for almost all $t \in [0, T]$ (see Grass et al. 2008, Theorem 3.4). The variable p is referred to as the *adjoint* or *costate* variable.

For a given initial value $x_0 \in \mathbb{R}^n$, the numerical approach now consists of finding functions $x : [0, T] \to \mathbb{R}^n$, $u : [0, T] \to U$ and $p : [0, T] \to \mathbb{R}^n$ satisfying

$$\dot{x}(t) = f(x(t), u(t)) \quad (11)$$

$$\dot{p}(t) = \delta p(t) - \mathcal{H}_x(x(t), p(t), u(t)) \quad (12)$$

$$u(t) = \underset{u \in U}{\mathrm{argmin}}\, \mathcal{H}(x(t), p(t), u) \quad (13)$$

$$x(0) = x_0, \quad p(T) = F_x(x(T)) \quad (14)$$

for $t \in [0, T]$. Depending on the regularity of the underlying data, the conditions (11)–(14) may only be necessary but not sufficient for x and u being an optimal trajectory $x^\star$ and control function $u^\star$, respectively. However usually x and u satisfying these conditions, are good candidates for the optimal trajectory and control, thus justifying the use of these conditions for the numerical approach. If needed, optimality of the candidates can be checked using suitable sufficient optimality conditions for which we refer to, e.g., Maurer (1981) or Malanowski et al. (2004). Due to the fact that in the maximum principle approach first optimality conditions are derived which are then discretized for numerical simulation, it is also termed *first optimize then discretize*.

Solving (11)–(14) numerically amounts to solving a boundary value problem, because the condition $x^\star(0) = x_0$ is posed at the beginning of the time interval $[0, T]$ while the condition $p(T) = F_x(x^\star(T))$ is required at the end. In order to solve such a problem, the simplest approach is the *single shooting* method which proceeds as follows:

We select a numerical scheme for solving the ordinary differential equations (11) and (12) for $t \in [0, T]$ with initial conditions $x(0) = x_0$,

$p(0) = p_0$ and control function $u(t)$. Then, we proceed iteratively as follows:

(0) Find initial guesses $p_0^0 \in \mathbb{R}^n$ and $u^0(t)$ for the initial costate and the control, fix $\varepsilon > 0$, and set $k := 0$.

(1) Solve (11) and (12) numerically with initial values x_0 and p_0^k and control function u^k. Denote the resulting trajectories by $\tilde{x}^k(t)$ and $\tilde{p}^k(t)$.

(2) Apply one step of an iterative method for solving the zero-finding problem $G(p) = 0$ with

$$G(p_0^k) := \tilde{p}^k(T) - F_x(\tilde{x}^k(T))$$

for computing p_0^{k+1}. For instance, in case of the Newton method we get

$$p_0^{k+1} := p_0^k - DG(p_0^k)^{-1} G(p_0^k).$$

If $\|p_0^{k+1} - p_0^k\| < \varepsilon$, stop; else compute

$$u^{k+1}(t) := \underset{u \in U}{\operatorname{argmin}} \mathcal{H}(x^k(t), p^k(t), u),$$

set $k := k + 1$, and go to (1).

The procedure described in this algorithm is called *single shooting* because the iteration is performed on the single initial value p_0^k. For an implementable scheme, several details still need to be made precise, e.g., how to parameterize the function $u(t)$ (e.g., piecewise constant, piecewise linear or polynomial), how to compute the derivative DG and its inverse (or an approximation thereof), and the argmin in (2). The last task considerably simplifies if the structure of the optimal control, e.g., the number of switchings in case of a bang-bang control, is known.

However, even if all these points are settled, the set of initial guesses p_0^0 and u^0 for which the method is going to converge to a solution of (11)–(14) tends to be very small. One reason for this is that the solutions of (11) and (12) typically depend very sensitively on p_0^0 and u^0. In order to circumvent this problem, *multiple shooting* can be used. To this end, one selects a time grid $0 = t_0 < t_1 < t_2 <$

$\ldots < t_N = T$ and in addition to p_0^k introduces variables $x_1^k, \ldots, x_{N-1}^k, p_1^k, \ldots, p_{N-1}^k \in \mathbb{R}^n$. Then, starting from initial guesses p_0^0, u^0, and $x_1^0, \ldots, x_{N-1}^0, p_1^0, \ldots, p_{N-1}^0$, in each iteration the Eqs. (11)–(14) are solved numerically on the intervals $[t_j, t_{j+1}]$ with initial values x_j^k and p_j^k, respectively. We denote the respective solutions in the k-th iteration by $\tilde{x}_j^k$ and $\tilde{p}_j^k$. In order to enforce that the trajectory pieces computed on the individual intervals $[t_j, t_{j+1}]$ fit together continuously, the map G is redefined as

$$G(x_1^k, \ldots, x_{N-1}^k, p_0^k, p_1^k, \ldots, p_{N-1}^k) =$$

$$\begin{pmatrix} \tilde{x}_0^k(t_1) - x_1^k \\ \vdots \\ \tilde{x}_{N-2}^k(t_1) - x_{N-1}^k \\ \tilde{p}_0^k(t_1) - p_1^k \\ \vdots \\ \tilde{p}_{N-2}^k(t_1) - p_{N-1}^k \\ \tilde{p}_{N-1}^k(T) - F_x(\tilde{x}_{N-1}^k(T)) \end{pmatrix}.$$

The benefit of this approach is that the solutions on the shortened time intervals depend much less sensitively on the initial values and the control, thus making the problem numerically much better conditioned. The obvious disadvantage is that the problem becomes larger as the function G is now defined on a much higher dimensional space but this additional effort usually pays off.

While the convergence behavior for the multiple shooting method is considerably better than for single shooting, it is still a difficult task to select good initial guesses x_j^0, p_j^0 and u^0. In order to accomplish this, homotopy methods can be used (see, e.g., Pesch 1994) or the result of a direct approach as presented in the next section can be used as an initial guess. The latter can be reasonable as the maximum principle-based approach can yield approximations of higher accuracy than the direct method.

In the presence of state constraints or mixed state and control constraints, the conditions (12)–(14) become considerably more technical and thus more difficult to be implemented numerically (cf. Pesch 1994).

Direct Discretization

Despite being the most straightforward and simple of the approaches described in this article, the direct discretization approach is currently the most widely used approach for computing single finite horizon optimal trajectories. In the direct approach, we first discretize the problem and then solve a finite dimensional nonlinear optimization problem (NLP), i.e., we *first discretize, then optimize*. The main reasons for the popularity of this approach are the simplicity with which constraints can be handled and the numerical efficiency due to the availability of fast and reliable NLP solvers.

The direct approach again applies to the finite horizon problem and computes an approximation to a single optimal trajectory $x^\star(t)$ and control function $u^\star(t)$ for a given initial value $x_0 \in X$. To this end, a time grid $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$ and a set $\mathcal{U}_d$ of control functions which are parameterized by finitely many values are selected. The simplest way to do so is to choose $u(t) \equiv u_j \in U$ for all $t \in [t_i, t_{i+1}]$. However, other approaches like piecewise linear or piecewise polynomial control functions are possible, too. We use a numerical algorithm for ordinary differential equations in order to approximately solve the initial value problems

$$\dot{x}(t) = f(x(t), u_i), \quad x(t_i) = x_i \qquad (15)$$

for $i = 0, \ldots, N - 1$ on $[t_i, t_{i+1}]$. We denote the exact and numerical solution of (15) by $x(t, t_i, x_i, u_i)$ and $\tilde{x}(t, t_i, x_i, u_i)$, respectively. Finally, we choose a numerical integration rule in order to compute an approximation

$$I(t_i, t_{i+1}, x_i, u_i) \approx \int_{t_i}^{t_{i+1}} e^{-\delta t}$$
$$g(x(t, t_i, x_i, u), u(t)) dt.$$

In the simplest case, one might choose $\tilde{x}$ as the Euler scheme and I as the rectangle rule, leading to

$$\tilde{x}(t_{i+1}, t_i, x_i, u_i) = x_i + (t_{i+1} - t_i) f(x_i, u_i)$$

and

$$I(t_i, t_{i+1}, x_i, u_i) = (t_{i+1} - t_i) e^{-\delta t_i} g(x_i, u_i).$$

Introducing the optimization variables $u_0, \ldots, u_{N-1} \in \mathbb{R}^m$ and $x_1, \ldots, x_N \in \mathbb{R}^n$, the discretized version of (2)–(4) reads

$$\underset{x_j \in \mathbb{R}^n, u_j \in \mathbb{R}^m}{\text{minimize}} \sum_{i=0}^{N-1} I(t_i, t_{i+1}, x_i, u) + e^{-\delta T} F(x_N)$$

subject to the constraints

$$
\begin{aligned}
u_j &\in U, & j &= 0, \ldots, N-1 \\
x_j &\in X, & j &= 1, \ldots, N \\
x_{j+1} &= \tilde{x}(t_{j+1}, t_j, x_j, u), & j &= 0, \ldots, N
\end{aligned}
$$

This way, we have converted the optimal control problem (2)–(4) into a finite dimensional nonlinear optimization problem (NLP). As such, it can be solved with any numerical method for solving such problems. Popular methods are, for instance, sequential quadratic programming (SQP) or interior point (IP) algorithms. The convergence of this approach was proved in Malanowski et al. (1998); for an up-to-date account on theory and practice of the method, see Gerdts (2012) and Betts (2010). These references also explain how information about the costates $p(t)$ can be extracted from a direct discretization, thus linking the approach to the maximum principle.

The direct method sketched here is again a multiple shooting method, and the benefit of this approach is the same as for solving boundary problems, thanks to the short intervals $[t_i, t_{i+1}]$; the solutions depend much less sensitively on the data than the solution on the whole interval $[0, T]$, thus making the iterative solution of the resulting discretized NLP much easier. The price to pay is again the increase of the number of optimization variables. However, due to the particular structure of the constraints guaranteeing continuity of the solution, the resulting matrices in the NLP have a particular structure which can be exploited numerically by a method called *condensing* (see Bock and Plitt 1984).

An alternative to multiple shooting methods are collocation methods, in which the internal variables of the numerical algorithm for solving (15) are also optimization variables. However, nowadays, the multiple shooting approach as described above is usually preferred. For a more detailed description of various direct approaches, see also Binder et al. (2001), Sect. 5.

Further Approaches for Infinite Horizon Problems

The last two approaches only apply to finite horizon problems. While the maximum principle approach can be generalized to infinite horizon problems, the necessary conditions become weaker and the numerical solution becomes considerably more involved (see Grass et al. 2008). Both the maximum principle and the direct approach can, however, be applied in a receding horizon fashion, in which an infinite horizon problem is approximated by the iterative solution of finite horizon problems. The resulting control technique is known under the name of model predictive control (MPC; see Grüne and Pannek 2011), and under suitable assumptions, a rigorous approximation result can be established.

Summary and Future Directions

The three main numerical approaches to optimal control are:

- The Hamilton-Jacobi-Bellman approach, which provides a global solution in feedback form but is computationally expensive for higher dimensional systems
- The Pontryagin maximum principle approach which computes single optimal trajectories with high accuracy but needs good initial guesses for the iteration
- The direct approach which also computes single optimal trajectories but is less demanding in terms of the initial guesses at the expense of a somewhat lower accuracy

Currently, the main trends in numerical optimal control lie in the areas of Hamilton-Jacobi-

Bellman equations and direct discretization. For the former, the development of discretization schemes suitable for increasingly higher dimensional problems is in the focus. For the latter, the popularity of these methods in online applications like MPC triggers continuing effort to make this approach faster and more reliable.

Beyond ordinary differential equations, the development of numerical algorithms for the optimal control of partial differential equations (PDEs) has attracted considerable attention during the last years. While many of these methods are still restricted to linear systems, in the near future we can expect to see many extensions to (classes of) nonlinear PDEs. It is worth noting that for PDEs, maximum principle-like approaches are more popular than for ordinary differential equations.

Cross-References

- ▶ Discrete Optimal Control
- ▶ Economic Model Predictive Control
- ▶ Nominal Model-Predictive Control
- ▶ Optimal Control and the Dynamic Programming Principle
- ▶ Optimal Control and Pontryagin's Maximum Principle
- ▶ Optimization Algorithms for Model Predictive Control

Bibliography

Bardi M, Capuzzo Dolcetta I (1997) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston

Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia

Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder JP, von Stryk O (2001) Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems: state of the art. Springer, Heidelberg, pp 295–340

Bock HG, Plitt K (1984) A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the 9th IFAC world congress, Budapest. Pergamon, Oxford, pp 242–247

Bryson AE, Ho YC (1975) Applied optimal control. Hemisphere Publishing Corp., Washington, DC. Revised printing

Falcone M (1997) Numerical solution of dynamic programming equations. In: Appendix A in Bardi M, Capuzzo Dolcetta I (eds) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston

Falcone M, Ferretti R (2013) Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations. SIAM, Philadelphia

Gerdts M (2012) Optimal control of ODEs and DAEs. De Gruyter textbook. Walter de Gruyter & Co., Berlin

Grass D, Caulkins JP, Feichtinger G, Tragler G, Behrens DA (2008) Optimal control of nonlinear processes. Springer, Berlin

Grüne L, Pannek J (2011) Nonlinear model predictive control: theory and algorithms. Springer, London

Malanowski K, Büskens C, Maurer H (1998) Convergence of approximations to nonlinear optimal control problems. In: Fiacco AV (ed) Mathematical programming with data perturbations. Lecture notes in pure and applied mathematics, vol 195. Dekker, New York, pp 253–284

Malanowski K, Maurer H, Pickenhain S (2004) Second-order sufficient conditions for state-constrained optimal control problems. J Optim Theory Appl 123(3):595–617

Maurer H (1981) First and second order sufficient optimality conditions in mathematical programming and optimal control. Math Program Stud 14: 163–177

McEneaney WM (2006) Max-plus methods for nonlinear control and estimation. Systems & control: foundations & applications. Birkhäuser, Boston

Pesch HJ (1994) A practical guide to the solution of real-life optimal control problems. Control Cybern 23(1–2):7–60

Vinter R (2000) Optimal control. Systems & control: foundations & applications. Birkhäuser, Boston

O

Observer-Based Control

H.L. Trentelman[1] and Panos J. Antsaklis[2]
[1]Johann Bernoulli Institute for Mathematics and
Computer Science, University of Groningen,
Groningen, AV, The Netherlands
[2]Department of Electrical Engineering,
University of Notre Dame, Notre Dame,
IN, USA

Abstract

An observer-based controller is a dynamic
feedback controller with a two-stage structure.
First, the controller generates an estimate of the
state variable of the system to be controlled,
using the measured output and known input
of the system. This estimate is generated by a
state observer for the system. Next, the state
estimate is treated as if it were equal to the exact
state of the system, and it is used by a static
state feedback controller. Dynamic feedback
controllers with this two-stage structure appear
in various control synthesis problems for linear
systems. In this entry, we explain observer-based
control in the context of internal stabilization by
dynamic measurement feedback.

Keywords

Detectability; Dynamic output feedback control;
Internal stabilization; Separation principle;
Stabilizability; State observers; Static state
feedback

Introduction

In this entry, we explain the notion of observer-
based feedback control. Given a to-be-controlled
system in input-state-output form, together with
a control objective, the problem is to design a
feedback controller such that the closed-loop
system meets the objective. In the case when
all state variables of the system are available
for control, the design problem is considered
to be simpler, and often the controller can be
chosen to be a static state feedback control law.
In the more general case where the controller
has access only to a linear function of the
state variables, the problem is more involved
and requires the design of a dynamic feedback
control law. The key idea of observer-based
feedback control is the following. As a first
step, one determines a state observer for the
system, i.e., a system that estimates the state of
the system based on the measured outputs and
inputs of the system. Next, the state estimate
is treated as if it were exactly equal to the
actual state of the system and is used by a
static state feedback controller. In this way, a
dynamic feedback controller is obtained that is
composed of a (dynamic) state observer and a
static feedback part.

Dynamic Output Feedback Control

Consider the controlled and observed system Σ:

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) + Ed(t), \\
y(t) &= Cx(t), \\
z(t) &= Hx(t),
\end{aligned} \tag{1}$$

with $x(t) \in \mathcal{X} = \mathbb{R}^n$ the state, $u(t) \in \mathbb{R}^m$ the control input, and $y(t) \in \mathbb{R}^p$ the measured output. The signal $d(t)$ may represent a disturbance input or a desired reference signal, while the signal $z(t)$ is a controlled output signal. A, B, C, E, and H are maps (or matrices). In general, a linear controller for this system is a finite-dimensional linear time-invariant system Γ represented by

$$\begin{aligned}
\dot{w}(t) &= Kw(t) + Ly(t), \\
u(t) &= Mw(t) + Ny(t).
\end{aligned} \tag{2}$$

The state space of the controller is assumed to be $\mathcal{W} = \mathbb{R}^q$ for some positive integer q. K, L, M, and N are assumed to be linear maps (or matrices). The controller (2) takes the observations y as its input and generates the control function u as its output. The closed-loop system resulting from the interconnection of Σ and Γ is described by the equations

$$\begin{aligned}
\begin{pmatrix} \dot{x}(t) \\ \dot{w}(t) \end{pmatrix} &= \begin{pmatrix} A+BNC & BM \\ LC & K \end{pmatrix} \begin{pmatrix} x(t) \\ w(t) \end{pmatrix} + \begin{pmatrix} E \\ 0 \end{pmatrix} d(t), \\
z(t) &= \begin{pmatrix} H & 0 \end{pmatrix} \begin{pmatrix} x(t) \\ z(t) \end{pmatrix}.
\end{aligned} \tag{3}$$

The control action of interconnecting the controller Γ with the system (1) is called *dynamic feedback*. The state space of the closed-loop system (3) is called the *extended state space* and is equal to the Cartesian product $\mathcal{X} \times \mathcal{W} = \mathbb{R}^{n+q}$. In general, a feedback control problem amounts to finding linear maps K, L, M, and N such that the closed-loop system (3) satisfies the control design specifications.

Observer-Based Controllers

Given the system (1) and a control objective, the problem thus arises on how to determine the maps K, L, M, and N so that the closed-loop systems meet the objective. As an example, take the special case when E in (1) is equal to zero (i.e., the system has no external disturbances or reference signals) and that we wish the closed-loop system (3) to be internally stable, i.e., we want to find the maps K, L, M, and N so that the eigenvalues λ_i of the system map of (3) are in the open left half-plane, i.e., satisfy $\text{Re}(\lambda_i) < 0$ for all i. If we had access to the entire state variable x (instead of only to the linear function $y = Cx$), then this problem would be simpler: assuming that the system is stabilizable (The system $\dot{x} = Ax + Bu$ is called stabilizable if there exists a map F such that $A + BF$ has all its eigenvalues in the open left half-plane), find a map F such that the eigenvalues of $A + BF$ are in the open left half-plane; then take the static state feedback controller $u = Fx$ as the control law. That is, we would choose the state space dimension of the controller Γ equal to 0 and the maps K, L, and M to be void, and we would take $N = F$.

In general, however, we only have access to a given linear function $y = Cx$ of x, determined by the output map C. The key idea of observer-based control is the following:

Use the theory of observer design to find an observer for the state x of the system (1), i.e., an observer that generates an estimate ξ of the system state x based on the measured output y and the control input u. Next, apply a static feedback $u = F\xi$ mimicking the (not permissible) control law $u = Fx$.

This idea leads to a dynamic feedback controller (2) of a very particular structure: the controller is the combination of a state observer (with a certain state space dimension) and a static control law acting on the state estimate. This *two-stage* structure, separating estimation and control, is often called the *separation principle*. We will work out this idea in more detail for the case when $E = 0$ (no external disturbances or reference signals) and the aim is to obtain internal

stability of the system. Before doing this, we first explain the most important material on observers that is needed in the sequel.

State Observers

If the state is not available for measurement, one can try to reconstruct it using a system, called observer, that takes the control input and the measured output of the original system as inputs and yields an output that is an estimate of the state of the original system. Again in case that in the system (1) we have $E = 0$, i.e., there are no disturbance signals. This is illustrated in the following picture:
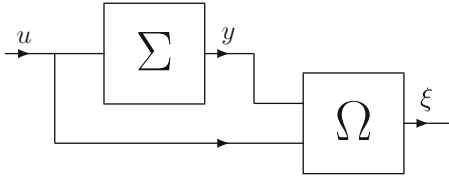
The quantity ξ is supposed to be an estimate, in some sense, of the state, and w is the state variable of the observer. In general, the observer, denoted by Ω, has equations of the form

$$\begin{aligned} \dot{w}(t) &= Pw(t) + Qu(t) + Ry(t), \\ \xi(t) &= Sw(t). \end{aligned} \quad (4)$$

It turns out that particular choices for P, Q, R, and S, specifically $P = A - GC$ (where the map G has to be determined), $Q = B$, $R = G$, and $S = I$, lead to

$$\dot{\xi}(t) = (A - GC)\xi(t) + Bu(t) + Gy(t). \quad (5)$$

Introducing the *estimation error* $e := \xi - x$ and interconnecting the system (1) with (5), we find that the error e satisfies the differential equation

$$\dot{e}(t) = (A - GC)e(t). \quad (6)$$

Hence all possible errors converge to 0 as t tends to infinity if and only if $A - GC$ is a stability matrix, i.e., has all its eigenvalues in the open left half-plane. In that case, we call (5) a *stable state observer*. Thus, a stable state observer exists if and only if G can be found such that $A - GC$ is a stability matrix. The problem of finding such a G is dual to the problem of finding a matrix F to a pair (A, B) such that $A + BF$ is a stability matrix.

Definition 1 The pair (C, A) is called *detectable* if there exists a matrix G such that $A - GC$ is a stability matrix, i.e., has all its eigenvalues in the open left half-plane.

Theorem 1 *Given system Σ, the following statements are equivalent*:
1. *Σ has a stable state observer.*
2. *(C, A) is detectable.*

The equation for ξ can be rewritten using an artificial output $\eta = C\xi$ as $\dot{\xi} = A\xi + Bu + G(y - \eta)$. The interpretation of this is as follows. If ξ is the exact state, then $\eta = y$, and hence ξ obeys exactly the same differential equation as x. Otherwise, the equation for ξ has to be corrected by a term determined by the *output error $y - \eta$*. Consequently, the state observer consists of an exact replica Σ_{dup} of the original system with an extra input channel for incorporating the output error and an extra output, the state of the observer, which serves as the desired estimate for the state of the original system. The following diagram depicts the situation:
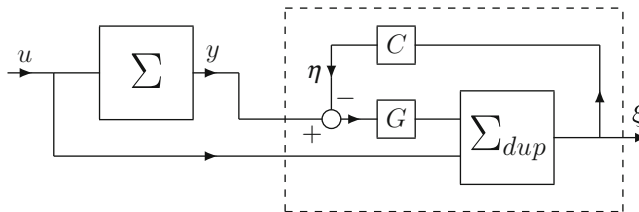
Observer-Based Stabilization

We now work out the ideas put forward in the previous sections for the special case of stabilization by dynamic measurement feedback, i.e., to find a controller (2) such that the closed-loop system (3) is internally stable; equivalently, the system mapping of (3) is a stability matrix. Again, we restrict ourselves to the case when $E = 0$.

We assume that we know how to stabilize by state feedback and how to build a state observer. If we have a plant of which we do not have the state available for measurement, we use a state observer to obtain an estimate of the state, and we apply the state feedback to this estimate rather than to the true state. This is illustrated by the following picture:
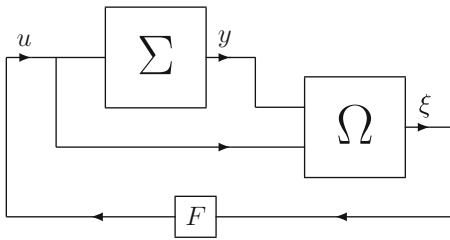
Again, consider the system Σ given by (1) and let the observer Ω be given by (5) . Combining this with $u = F\xi$ yields

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + BF\xi, \\
\dot{\xi}(t) &= (A - GC + BF)\xi(t) + GCx(t).
\end{aligned}
\tag{7}
$$

Introducing again $e := \xi - x$, we obtain, in accordance with the previous section,

$$
\begin{aligned}
\dot{x}(t) &= (A + BF)x(t) + BFe(t), \\
\dot{e}(t) &= (A - GC)e(t).
\end{aligned}
$$

That is, the equation $\dot{x}_e = A_e x_e$ with

$$
x_e := \begin{pmatrix} x \\ e \end{pmatrix}, \quad A_e := \begin{pmatrix} A + BF & BF \\ 0 & A - GC \end{pmatrix}.
$$

Assume that Σ is stabilizable and detectable. Then F and G can be found such that $A + BF$ and $A - GC$ are stability matrices. Since the set of eigenvalues of A_e is the union of those of $A + BF$ and $A - GC$, it follows that A_e is a stability matrix. Consequently, the system $\dot{x}_e = A_e x_e$ is asymptotically stable; equivalently, every solution $x_e = (x, e)$ converges to 0 as t tends to infinity. Of course, if (x, ξ) is a solution of (7), then $\xi = x + e$, with $x_e = (x, e)$ a solution of $\dot{x}_e = A_e x_e$. Hence (x, ξ) also converges to 0 as t goes to infinity. Thus we have proved the "if" part of the following theorem:

Theorem 2 *There exists an internally stabilizing dynamic feedback controller for Σ if and only if Σ is stabilizable and detectable. A controller is given by*

$$
\begin{aligned}
\dot{\xi}(t) &= (A - GC)\xi(t) + Bu(t) + Gy(t), \\
u(t) &= F\xi(t),
\end{aligned}
\tag{8}
$$

where F is any map such that $A + BF$ is a stability matrix and G is any map such that $A - GC$ is a stability matrix.

The controller (8) is an *observer-based dynamic feedback controller*, since it is composed of a state observer and a static feedback part.

Summary and Future Directions

We have given an introduction to observer-based feedback controllers and have explained that such controllers are dynamic feedback controllers that can be represented as the composition of a state observer for the system, together with a static control law mimicking a (not permitted) static state feedback control law. We have given a detailed description of this principle for the case that the system to be controlled has no external disturbances or reference signals and the control objective is internal stability of the system. More intricate versions of the principle of observer-based feedback control appear in control design problems for linear systems with external disturbances and reference signals and with different, more sophisticated, control objectives. Examples

of these are the regulator problem, the problem of disturbance decoupling with internal stability, the $\mathcal{H}_2$ optimal control problem, and the $\mathcal{H}_\infty$ suboptimal control problem.

Cross-References

▶ Linear State Feedback
▶ Observers in Linear Systems Theory

Bibliography

Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston

Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London

Wonham WM (1979) Linear multivariable control: a geometric approach. Springer, New York

Zadeh LA, Desoer CA (1963) Linear systems theory – the state-space approach. McGraw-Hill, New York

Observers for Nonlinear Systems

Laurent Praly
MINES ParisTech, PSL Research University, CAS, Fontainebleau, France

Abstract

Observers are objects delivering estimation of variables which cannot be directly measured. The access to such *hidden* variables is made possible by combining modeling and measurements. But this is bringing face to face real world and its abstraction with, as a result, the need for dealing with uncertainties and approximations leading to difficulties in implementation and convergence.

Keywords

Introduction

Observers are answers to the question of estimating, from observed/measured/empirical variables, denoted y, and delivered by sensors equipping a real-world system, some "theoretical" variables, called hidden variables in this text, denoted z, which are involved in a mathematical model related to this system. The measured variables make what is called the a posteriori information on the hidden variables, whereas the model is part of the a priori information. Because a model cannot fit exactly a system, introduction of uncertainties is mandatory.

Typically this model describing the link between hidden and measured variables is made of three components:

- *A dynamic model* describes the dynamics/evolution ($\dot{x}$ denotes the time derivative $\frac{dx}{dt}$):

$$\dot{x}(t) = f(x(t), t, \delta^s(t)) \text{ resp. } x_{k+1} = f_k\left(x_k, \delta^s_k\right), \tag{1}$$

where t, in the continuous case, or k, in the discrete case, is an evolution parameter, called time in this text; x is a state, assumed finite dimensional in this text; and δ^s represents the uncertainties in the state dynamics. Any possible known inputs are represented here by the time dependence of f.

- *A sensor model* relates state and measured variables:

$$y(t) = h\left(x(t), t, \delta^m(t)\right) \text{ resp. } y_k = h_k\left(x_k, \delta^m_k\right) \tag{2}$$

with δ^m representing the uncertainties in the measurements.

- A model which relates state and hidden variables:

$$z(t) = \mathcal{U}\left(x, t, \delta^h(t)\right) \text{ resp. } z_k = \mathcal{U}_k\left(x_k, \delta^h_k\right) \tag{3}$$

where again δ^h represents the uncertainties in the hidden variables.

In a deterministic setting, the a priori information on the uncertainties $(\delta^s, \delta^m, \delta^h)$ may be that the values of δ^s, δ^m, and δ^h are unknown but belong to known sets Δ^s, Δ^m, and Δ^h. Namely, we have:

O

$$\delta^s(t) \in \Delta^s(t), \; \delta^m(t) \in \Delta^m(t), \delta^h(t) \in \Delta^h(t) \,,$$
$$\text{respectively}, \delta_k^s \in \Delta_k^s \,, \; \delta_k^m \in \Delta_k^m \,, \; \delta_k^h \in \Delta_k^h \,. \tag{4}$$

In a stochastic setting and more specifically in a Bayesian approach, it may be that δ^s, δ^m, and δ^h are unknown realizations of stochastic processes for which we know the probability distributions.

Similarly we may also know a priori that we have:

$$\begin{aligned} & x(t) \in \mathcal{X}(t), && z(t) \in \mathcal{Z}(t) \\ \text{respectively}, \quad & x_k \in \mathcal{X}_k, && z_k \in \mathcal{Z}_k \end{aligned} \tag{5}$$

where the sets $\mathcal{X}$ and $\mathcal{Z}$ are known or we may have a priori probability distribution for x and z.

In this context, the a priori information is the data of the functions f, h, and $\mathcal{Y}$, of the sets Δ^s, Δ^m, and Δ^h or the corresponding probability distribution and so may be also of the sets $\mathcal{X}$ and $\mathcal{Z}$ or the corresponding a priori probability distribution.

In the next section, we state the observation problem and give the solutions which are direct consequences of the deterministic and stochastic setting given above. This will allow us to see that an observer is actually a dynamical system with the measurements as inputs and the estimate as output. But approximations in the implementation of these solutions, not knowing how to initialize, may lead to convergence problems even when the uncertainties disappear. The second part of this text is devoted to this convergence topic.

To ease the presentation, we deal only with the discrete time case in section "Set Valued and Conditional Probability Valued Observers" and the continuous time case in sections "An Optimization Approach" and "Convergent Observers."

Observation Problem and Its Solutions

The Observation Problem

Let $X^{\delta^s}(x, t, s)$, respectively $X_l^{\delta^s}(x, k)$, denote a solution of (1) at time s, respectively l, going through x at time t, respectively k, and under the action of δ^s.

Observation problem *At each time t, respectively k, given the function $s \in]t - T, t] \mapsto y(s)$, respectively the sequence $l \in \{k - K, \dots, k\} \mapsto y_l$, find an estimation $\hat{z}(t)$, respectively $\hat{z}_k$, of $z(t)$, respectively, z_k, satisfying*

$$\hat{z}(t) = \mathcal{Y}\left(\hat{x}(t), t, \delta^h(t)\right) \; resp. \; \hat{z}_k = \mathcal{Y}_k\left(\hat{x}_k, \delta_k^h\right) \,.$$

where $\hat{x}(t)$, respectively $\hat{x}_k$, is to be found as a solution of

$$\hat{x}(t) \in \mathcal{X}(t) \,,$$
$$y(s) = h\left(X^{\delta^s}(\hat{x}(t), t, s), s, \delta^m(s)\right) \; \forall s \in]t - T, t],$$

respectively

$$\hat{x}_k \in \mathcal{X}_k \,,$$
$$y_l = h_l\left(X_l^{\delta^s}(\hat{x}_k, k), \delta_l^m\right) \; \forall l \in \{k - K, \dots, k\}$$

and where the time functions δ^s, δ^m, and δ^h must agree with the a priori (deterministic/stochastic) information or minimized in some way.

In this statement T, respectively K, quantifies the time window length or memory length during which we record the measurement. The accumulation with time of measurements, together with the model equations (1)–(3) and the assumptions on $(\delta^s, \delta^m, \delta^h)$, gives a redundancy of data compared with the number of unknowns that the hidden variables are. This is why it may be possible to solve this observation problem.

To simplify the following presentation, we restrict our attention on the case where the hidden variables are actually the full model state, i.e.,

$$z = \mathcal{Y}(x) = x \,.$$

When z differs from x, observers are called functional observers.

Set-Valued and Conditional Probability-Valued Observers

Conceptually the answer to this problem is easy at least when the memory increases with time ($\dot{T}(t) = 1$ resp. $K_{k+1} = K_k + 1$) leading to an infinite non-fading memory. It consists in starting

from all what the a priori information makes possible and to eliminate what is not consistent with the a posteriori information. In the set-valued observer setting, in the discrete time case, this gives the following observer. To ease its reading, we underline the data given by the a priori information. It requires the introduction of two sets ξ_k and $\xi_{k|k-1}$ which are updated at each time k when a new measurement y_k is made available. ξ_k is the set which x_k is guaranteed to belong to at time k, knowing all the measurements up to time k, and $\xi_{k|k-1}$ is the same but with measurements known up to time $k-1$.

Set-valued observer:

Initialization: $\quad\quad\quad \xi_0 = \underline{\mathcal{X}_0}$

At each time k: prediction (flowing) $\quad \xi_{k|k-1} = \underline{f_{k-1}}(\xi_{k-1}, \underline{\Delta^s_{k-1}})$

restriction (consistency) $\quad \xi_k = \big\{x \in \big(\xi_{k|k-1} \bigcap \underline{\mathcal{X}_k}\big) :$

$$y_k \in \underline{h_k}(x, \underline{\Delta^m_k})\big\}$$

estimation $\quad\quad\quad\quad \hat{x}_k \in \xi_k$

A key feature here is that *this observer has a state ξ_k – a set – and is a dynamical system in the form:*

$$\xi_{k+1} = \varphi_k(\xi_k, y_k), \quad\quad \hat{x}_k \in \xi_k$$

with y as input and $\hat{x}$ as output which is not single valued. Important also, the initial condition of the state ξ is given by the a priori information.

In the stochastic setting, following the Bayesian paradigm, the observer has the same structure but with the state ξ_k being a conditional probability. See Jazwinski (2007, Theorem 6.4) or Candy (2009, Table 2.1). In that setting too the observer is not a single state; it is the (a posteriori) conditional probability of the random variable x_k given the a priori information and the sequence of measurements $l \in \{k-K, \ldots, k\} \mapsto y_l$.

Comments

Implementation: For the time being, except for very specific cases (Kalman filter, . . .), the set-valued and the conditional probability-valued observers remain conceptual since we do not know how to manipulate numerically sets and probability laws. Their implementation requires approximations. For instance, see

Milanese et al. (1996) and Witsenhausen (1966) for the set case and Arulampalam et al. (2002), Bucy and Joseph (1987), Candy (2009), and Jazwinski (2007) for the conditional probability case.

Need of finite or infinite but fading memory: In these observers, model states x which are consistent with the a priori information but do not agree with the a posteriori information are eliminated (set intersection or probability product). But once a point is eliminated, this is forever. As a consequence if there is, at some time, a misfit between a priori and a posteriori information, it is mistakenly propagated in future times. A way to round this problem is to keep the information memory finite or infinite but fading. In particular, with fixed length memory, consistent points which were disregarded due to measurements which are no more in the memory are reintroduced. This says also that observers should not be sensitive to their initial condition.

Not single-valued estimate. The observers introduced above realize a lossless data compression with extracting and preserving all what concerns the hidden variables in the redundant data given by a priori and a posteriori information. But this "lossless compression" answer is not single valued (set valued or conditional probability valued) as a result of taking uncertainties into account. Actually, to get a single-valued answer, *the observation problem must be complemented by making precise for what the estimation is made.* For instance, we may want to select the most likely or the average or more generally some cost-minimizing estimate $\hat{x}$ among all the possible ones given by ξ. In this way we obtain an observer giving a single-valued estimate:

$$\xi_{k+1} = \varphi_k(\xi_k, y_k), \quad\quad \hat{x}_k = \tau_k(\xi_k)$$

respectively

$$\dot{\xi}(t) = \varphi(\xi(t), y(t), t), \quad \hat{x}(t) = \tau(\xi(t), t) \tag{6}$$

But then, in general, we lose information, and in particular we have no idea on the confidence

level this estimate has. Also, since the function τ, at least, encodes for what the estimate $\hat{x}$ is used, for different uses, different functions τ may be needed.

An Optimization Approach

A shortcut to obtain directly an observer giving a single-valued estimate is to design it by trading off among a priori and a posteriori information (see Cox 1964, pages 7–10; Alamir 2007). For example, in the continuous time case, we can select the estimate $\hat{x}(t)$ among the minimizers (in x) of

$$\mathcal{C}(\{s \mapsto \delta^s(s)\}, x, t) = \int_{-\infty}^{t} C\left(\delta^s(s), y(s),\right.$$
$$\left. X^{\delta^s}(x, t, s), s\right) ds$$

where $X^{\delta^s}(x, t, s)$ is still the notation for a solution to (1) and $\{s \mapsto \delta^s(s)\}$, representing the unmodelled effect on the dynamics, is among the arguments for the minimization. The infinitesimal cost C is chosen to take nonnegative values and be such that $C(0, h(x, s), x, s)$ is zero. For instance, it can be

$$C(\delta^s, y, x, s) = \|\delta^s\|_x^2 + d_y(y, h(x, s))^2$$

where $\|.\|_x$ is a norm at the point x and d_y is a distance in the measurement space. In the same spirit, instead of optimization, a minimax approach can be followed. See, for instance, Bertsekas and Rhodes (1971), Başar and Bernhard (1995, Chapter 7), and Willems (2004).

With x fixed, the minimization of $\mathcal{C}$ is an infinite horizon optimal control problem in reverse time. Solving on line this problem is extremely difficult and again approximations are needed. We do not go on with this approach, but we remark that, under extra assumptions, the observer we obtain following this approach can also be implemented in the form of a dynamical system (6) but with the specificity that *the estimate $\hat{x}$ is part of the observer state ξ and its dynamics are a copy of the undisturbed model with a correction term which is zero when the estimated*

state reproduces the measurement. Namely, we get

$$\dot{\hat{x}}(t) = f(\hat{x}(t), t, 0) + E\Big(\{\sigma \mapsto y(\sigma)\}, \hat{x}(t), y(t), t\Big)$$

where E is zero when $h(\hat{x}(t), t) = y(t)$. But, as opposed to what we saw in the previous section, the initial condition for the part $\hat{x}$ of the observer state is unknown. Hence, we encounter again the need for the observer to forget its initial condition.

Convergent Observers

We have mentioned that often an observer can be implemented as a dynamical system, but without knowing necessarily how to initialize it. Also approximation is involved both in its design and its implementation. So, at least when it gives a single-valued estimate, we are facing the problem of convergence of this estimate to the "true" value, at least when there is no uncertainties. We concentrate now our attention on the study of this convergence, but, to simplify, in the continuous time case only.

Let the model and observer dynamics be

$$\dot{x}(t) = f(x(t), t), \quad y(t) = h(x(t), t) \quad (7)$$

$$\dot{\xi} = \varphi(\xi(t), y(t), t), \quad \hat{x}(t) = \tau(\xi(t), y(t), t) \tag{8}$$

with the observer state ξ of finite dimension m. We denote by $(X(x, t, s), \Xi((x, \xi), t, s))$ a solution of (7)–(8).

Since we are dealing with convergence, the focus is on what is going on when the time becomes very large and in particular on the set Ω of model states which are accumulation points of some solution. Specifically we are interested in the stability properties of the set

$$\mathfrak{Z}(t) = \Big\{(x, \xi) : x \in \Omega \,\&\, x = \tau(\xi, h(x, t), t)\Big\}$$

which is contained in the zero estimation error set associated with the given model-observer pair.

Definition 1 (convergent observer) We say the observer (8) is convergent if for each t, there

exists a set $\mathfrak{Z}_a(t) \subset \mathfrak{Z}(t)$, such that on the domain of existence of the solution, a distance between the point $(X(x,t,s), \Xi((x,\xi),t,s))$ and the set $\mathfrak{Z}_a(s)$ is upperbounded by a real function $s \mapsto \beta^c_{x,\xi,t}(s)$, may be dependent on (x,ξ,t), with nonnegative values, strictly decreasing and going to zero as s goes to infinity.

Necessary Conditions for Observer Convergence

No Restriction on τ

It is possible to prove that *if the observer is convergent, then,*

- *Necessity of detectability:* When h and τ are uniformly continuous in x and ξ, respectively, the estimate $\hat{x}$ does converge to the model state x. In this case, two solutions of the model (7) which produce the same measurement must converge to each other. This is an asymptotic distinguishability property called detectability. If we are interested not only in the asymptotic behavior but also in the transient (as for output feedback), a property stronger than detectability is needed. In particular instantaneous distinguishability (see section "Observers Based on Instantaneous Distinguishability") is necessary if we want to be able to impose the decay rate of the function $\beta^c_{x,\xi,t}$.

- *Necessity of $m \geq n - p$:* For each t, there exists a subset $\mathcal{X}_a(t)$ of Ω, supposed to collect the model states which can be asymptotically estimated and such that we can associate, to each of its point x, a set $\tau^i(x,t)$ allowing us to redefine the set $\mathfrak{Z}_a(t)$ as

$$\mathfrak{Z}_a(t) = \left\{ (x,\xi) : \ x \in \mathcal{X}_a(t) \ \& \ \xi \in \tau^i(x,t) \right\} .$$

This implies that for each t and each x in $\mathcal{X}_a(t)$, there is a point ξ satisfying

$$x = \tau(\xi, h(x,t), t) . \tag{9}$$

This is a surjectivity property of the function τ but of a special kind since $h(x,t)$ is an argument of τ. We say that, for each t, *the function τ is surjective to $\mathcal{X}_a(t)$ given h.* In a "generic" situation this property requires the dimension m of the observer state ξ to be larger or equal to the dimension n of the model state x minus the dimension p of the measurement y.

τ Is Injective Given h

We consider now the case where the observer has been designed with a function τ which is injective given h, namely, we have the following implication, when x is in $\mathcal{X}_a(t)$,

$$\Big[\tau(\xi_1, h(x,t), t) = \tau(\xi_2, h(x,t), t)$$
$$\& \ \xi_1 \in \tau^i(x,t) \Big]$$
$$\implies \quad \xi_1 = \xi_2 .$$

In a "generic" situation, this property, together with the surjectivity given h, implies that the dimension m of the observer state ξ should be between $n - p$ and n.

If a convergent observer has such a function τ, then $(x,t) \mapsto \tau^i(x,t)$, which is (of course) a (single valued) function, admits a Lie derivative $(L_f \tau^i(x,t) = \lim_{dt \to 0} \frac{\tau^i(X(x,t,t+dt),t+dt) - \tau^i(x,t)}{dt})$ $L_f \tau^i$ satisfying

$$L_f \tau^i(x,t) = \varphi(\tau^i(x,t), h(x,t), t) \ \forall x \in \mathcal{X}_a(t) \tag{10}$$

This says (very approximatively) that φ is nothing but the image of the vector field f, under the change of coordinates $(x,t) \mapsto (\tau^i(x,t), t)$ but again all this given h. As partly obtained in the optimization approach, the observer dynamics are then a copy of the model dynamics with maybe a correction term which is zero when the estimated state reproduce the measurement.

If moreover the functions h and τ are uniformly continuous in x and ξ, respectively, then, given ξ_1 and ξ_2 a distance between $\Xi((x,\xi_1),t,s)$ and $\Xi((x,\xi_2),t,s)$ goes to zero as s goes to infinity. This property is related to what was called extreme stability (see Yoshizawa 1966) in the 1950s and 1960s and is called incremental stability today (see Angeli 2002). It holds when, with denoting by $\Xi^y(\xi,t,s)$ the solution at time s of the observer dynamics :

O

$$\dot{\xi}(t) \; = \; \varphi(\xi(t), y(t), t)$$

going through ξ at time t and under the action of y, *the flow $\xi \; \mapsto \; \Xi^y(\xi, t, s)$ is a strict contraction (see Jouffroy (2005) for a bibliography on contraction) for each $s > t$ or, at least, if a distance between any two solutions $\Xi^y(\xi_1, t, s)$ and $\Xi^y(\xi_2, t, s)$, with the same input y, converges to* 0.

Sufficient Conditions

Knowing now how a convergent observer should look like, we move to a quick description of some such observers.

Observers Based on Contraction

Since the flow generated by the observer should be a contraction, we may start its design by picking the function φ as

$$\dot{\xi}(t) \; = \; \varphi(\xi(t), y(t), t) \; = \; A\,\xi(t) \, + \, B(y(t), t)$$

where A, not related to f, is a matrix whose eigenvalues have strictly negative real part. Under weak restriction, there exists a function τ^i satisfying (10), namely,

$$L_f \tau^i(x, t) \; = \; A\, \tau^i(x, t) \, + \, B(h(x, t), t) \,. \tag{11}$$

To obtain a convergent observer, it is then sufficient that there exists a (uniformly continuous) function τ satisfying

$$x \; = \; \tau(\tau^i(x, t), h(x, t), t)$$

For this to be possible, the function τ^i should be injective given h. This injectivity holds when the observer state has dimension $m \geq 2(n + 1)$, the model is distinguishable, and provided the eigenvalues of A have a sufficiently negative real part and are not in a set of zero Lebesgue measure.

Unfortunately, we are facing again a possible difficulty in the implementation since an expression for a function τ^i satisfying (11) is needed and the function $\tau : (\xi, y, t) \mapsto \hat{x}(t)$ is known implicitly only as

$$\xi \; = \; \tau^i(\hat{x}(t), t) \,.$$

See Andrieu and Praly (2006), Luenberger (1964), and Shoshitaishvili (1990).

Observers Based on Instantaneous Distinguishability

Instantaneous distinguishability means that we can distinguish as quickly as we want two model states by looking at the paths of the measurements they generate. A sufficient condition to have this property can be obtained by looking at the Taylor expansion in s of $h(X(x, t, s), s)$. Indeed, we have:

$$h(X(x, t, s), s) = \sum_{i=0}^{m-1} h_i(x, t) \frac{(s - t)^i}{i!}$$
$$+ o\left((s - t)^{m-1}\right)$$

where h_i is a function obtained recursively as

$$h_0(x, t) \; = \; h(x, t)$$
$$h_{i+1}(x, t) \; = \; \widehat{\dot{h_i(x, t)}} \; = \; \frac{\partial h_i}{\partial x}(x, t) f(x, t) \\ + \frac{\partial h_i}{\partial t}(x, t).$$

If there exists an integer m such that, in some uniform way with respect to t, the function

$$x \; \mapsto \; H_m(x, t) = (h_0(x, t), \, \ldots, \, h_{m-1}(x, t))$$

is injective, then we do have instantaneous distinguishability. We say the system is differentially observable of order m when this injectivity property holds. When a system has such a property, the model state space has a very specific structure as discussed in Isidori (1995, Section 1.9). It means that we can reconstruct x from the knowledge of y and its $m-1$ first time derivatives, i.e., there exists a function Φ such that we have:

$$x \; = \; \Phi\left(H_m(x, t), t\right).$$

This way, we are left with estimating the derivatives of y. This can be done as follows. With the notation $\eta_i \; = \; h_{i-1}(x, t)$, we obtain:

$$\dot{\eta}(t) \;=\; F\,\eta + G\,h_m\big(\Phi(\eta(t),t)\,,\,t\big)$$

where

$$F\,\eta = (\eta_2,\,\ldots\,,\eta_m,\,0),\ \ G = (0,\,\ldots\,,0,\,1).$$

When the last term on the right hand side is Lipschitz, we can find a convergent observer in the form:

$$\dot{\xi}(t) = F\,\xi(t) + G\,h_m\,(\hat{x}(t),t) + K(y(t) - \xi_1(t)),$$
$$\hat{x}(t) = \tau\,(\xi(t),t),$$

with ξ being actually an estimation of η and where K is a constant matrix and τ is a modified version of Φ keeping the estimated state in its a priori given set $\mathcal{X}(t)$.

This is the high-gain observer paradigm. See Gauthier and Kupka (2001) and Tornambe (1988). The implementation difficulty is in the function $\hat{\Phi}$, not to mention sensitivity to measurement uncertainty.

Observers with τ Bijective Given h

Case Where τ Is the Identity Function A convergent observer whose function τ is the identity has the following form:

$$\dot{\xi} = f(\xi,t)$$
$$+ E\left(\{\sigma \mapsto y(\sigma)\}, \xi(t), y(t), t\right), \hat{x}(t) = \xi(t).$$
$$(12)$$

The only piece remaining to be designed is the correction term E. It has to ensure convergence and may be also other properties like symmetry preserving (see Bonnabel et al. 2008).

For this design, a first step is to exhibit some specific properties of the vector field f by writing it in some appropriate coordinates. For example, there may exist coordinates such that the expression of f takes the form $\mathfrak{f}(x(t), h(x,t), t)$ and the corresponding observer (12) is such that there exists a positive definite matrix P for which the function $s \mapsto (X(x,t,d) - \hat{X}((x,\hat{x}),t,s))' P(X(x,t,d) - \hat{X}((x,\hat{x}),t,s))$ is strictly decaying (if not zero). A necessary condition for this to be possible is that $\mathfrak{f}$ is

monotonic tangentially to the level sets of the function h, i.e., for all (x,y,v,t) satisfying $y = h(x,t)$ and $\frac{\partial h}{\partial x}(x,t)v = 0$, we have:

$$v^T P\,\frac{\partial \mathfrak{f}}{\partial x}(x,y,t)\,v \;\leq\; 0. \qquad (13)$$

This is another way of expressing a detectability condition. This expression is coordinate dependent, hence the importance of choosing the coordinates properly.

When this condition is strict and uniform in t, it is sufficient to get a locally convergent observer and even a nonlocal one when h is linear in x, i.e., $h(x,t) = H(t)x$, again a coordinate-dependent condition. In this latter case the observer takes the form

$$\dot{\xi}(t) = \mathfrak{f}(\xi(t), y(t), t) \;+\; \ell(\xi(t))\,P^{-1}H(t)^T$$
$$[y(t) - H(t)\xi(t)],$$
$$\hat{x}(t) = \xi(t),$$

where ℓ is a real function to be chosen with sufficiently large values. If (13) is strict and uniform and holds for all v, the correction term is not needed.

There are many other results of this type, exploiting one or the other specificity of the dependence on x of the function $\mathfrak{f}$ – monotonicity, convexity, ... See Fan and Arcak (2003), Krener and Isidori (1983), Respondek et al. (2004), Sanfelice and Praly (2012), ...

Case Where $(x,t) \mapsto (\tau^i(x,t), h(x,t), t)$ Is a Diffeomorphism At each time t we know already that the model state x we want to estimate satisfy $y(t) = h(x,t)$. So, as remarked in Luenberger (1964), when $(h(x,t), t)$ can be used as part of coordinates for (x,t), we need to estimate the remaining part only. This can be done if we find a function τ^i, whose values are $n - p$ dimensional, such that $(x,t) \mapsto (y,\eta,t) = (h(x,t), \tau^i(x,t), t)$ is a diffeomorphism and the flow $\eta \mapsto \eta^y(\eta,t,s)$ generated by

$$\dot{\eta}(t) = \frac{\partial \tau^i}{\partial x}(x(t),t)f(x(t),t) \;+\; \frac{\partial \tau^i}{\partial t}(x(t),t),$$
$$= \varphi(\eta(t), y(t), t)$$

is a strict contraction for all $s > t$. Indeed in this case the observer dynamics can be chosen as

$$\dot{\xi}(t) \;=\; \varphi(\xi(t), y(t), t)$$

and the estimate $\hat{x}(t)$ is obtained as solution of

$$\tau^i(\hat{x}(t), t) \;=\; \xi(t), \qquad h(\hat{x}(t), t) \;=\; y(t).$$

This is the reduced-order observer paradigm. See, for instance, Besançon (2000, Proposition 3.2), Carnevale et al. (2008), and Luenberger (1964, Theorem 4).

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Observers in Linear Systems Theory
▶ Regulation and Tracking of Nonlinear Systems

Bibliography

Alamir M (2007) Nonlinear moving horizon observers: theory and real-time implementation. In: Nonlinear observers and applications. Lecture notes in control and information sciences. Springer, Berlin/New York

Andrieu V, Praly L (2006) On the existence of Kazantzis-Kravaris/Luenberger observers. SIAM J Control Optim 45(2):432–456

Angeli D (2002) A Lyapunov approach t incremental stability properties. IEEE Trans Autom Control 47(3):410–421

Arulampalam M, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188

Bain A, Crisan D (2009) Fundamentals of stochastic filtering. Stochastic modelling and applied probability, vol 60. Springer, New York/London

Başar T, Bernhard P (1995) H^∞ optimal control and related minimax design problems: a dynamic game approach, revised 2nd edn. Birkäuser, Boston

Bertsekas D, Rhodes IB (1971) On the minimax reachability of target sets and target tubes. Automatica 7:233–213

Besançon G (2000) Remarks on nonlinear adaptive observer design. Syst Control Lett 41(4):271–280

Bonnabel S, Martin P, Rouchon P (2008) Symmetry-preserving observers. IEEE Trans Autom Control 53(11):2514–2526

Bucy R, Joseph P (1987) Filtering for stochastic processes with applications to guidance, 2nd edn. Chelsea Publishing Company, Chelsea

Candy J (2009) Bayesian signal processing: classical, modern, and particle filtering methods. Wiley series in adaptive learning systems for signal processing, communications and control. Wiley, Hoboken

Carnevale D, Karagiannis D, Astolfi A (2008) Invariant manifold based reduced-order observer design for nonlinear systems. IEEE Trans Autom Control 53(11):2602–2614

Cox H (1964) On the estimation of state variables and parameters for noisy dynamic systems. IEEE Trans Autom Control 9(1):5–12

Fan X, Arcak M (2003) Observer design for systems with multivariable monotone nonlinearities. Syst Control Lett 50:319–330

Gauthier J-P, Kupka I (2001) Deterministic observation theory and applications. Cambridge University Press, Cambridge/New York

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York

Jazwinski A (2007) Stochastic processes and filtering theory. Dover, Mineola

Jouffroy J (2005) Some ancestors of contraction analysis. In: Proceedings of the IEEE conference on decision and control, Seville, pp 5450–5455

Krener A, Isidori A (1983) Linearization by output injection and nonlinear observer. Syst Control Lett 3(1):47–52

Luenberger D (1964) Observing the state of a linear system. IEEE Trans Mil Electron MIL-8:74–80

Milanese M, Norton J, Piet-Lahanier H, Walter E (eds) (1996) Bounding approaches to system identification. Plenum Press, New York

Respondek W, Pogromsky A, Nijmeijer H (2004) Time scaling for observer design with linearizable error dynamics. Automatica 40:277–285

Sanfelice R, Praly L (2012) Convergence of nonlinear observers on with a riemannian metric (Part I). IEEE Trans Autom Control 57(7):1709–1722

Shoshitaishvili A (1990) Singularies for projections of integral manifolds with applications to control and observation problems. In: Arnol'd VI (ed) Advances in Soviet mathematics. Theory of singularities and its applications, vol 1. American Mathematical Society, Providence

Tornambe A (1988) Use of asymptotic observers having high gains in the state and parameter estimation. In: Proceedings of the IEEE conference on decision and control, Austin, pp 1791–1794

Willems JC (2004) Deterministic least squares filtering. J Econ 118:341–373

Witsenhausen H (1966) Minimax control of uncertain systems. Elec. Syst. Lab. M.I.T. Rep. ESL-R-269M, Cambridge, May 1966

Yoshizawa T (1966) Stability theory by Lyapunov's second method. The Mathematical Society of Japan, Tokyo

Observers in Linear Systems Theory

A. Astolfi[1,2] and Panos J. Antsaklis[3]

[1]Department of Electrical and Electronic Engineering, Imperial College London, London, UK

[2]Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

[3]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA

Abstract

Observers are dynamical systems which process the input and output signals of a given dynamical system and deliver an online estimate of the internal state of the given system which asymptotically converges to the exact value of the state. For linear, finite-dimensional, time-invariant systems, observers can be designed provided a weak observability property, known as detectability, holds.

Keywords

Linear systems; Observers; Reduced order observer; State estimation

Introduction

Consider a linear, finite-dimensional, time-invariant system described by equations of the form

$$\sigma x = Ax + Bu,$$
$$y = Cx + Du, \tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ and A, B, C, and D matrices of appropriate dimensions and with constant entries, and the problem of estimating its state from measurements of the input and output signals. In Eq. (1) $\sigma x(t)$ stands for $\dot{x}(t)$, if the system is continuous-time, and for $x(t + 1)$, if the system is discrete-time. In addition, if the system is continuous-time, then $t \in \mathbb{R}^+$, i.e., the set of nonnegative real numbers, whereas if the system is discrete-time, then $t \in Z^+$, i.e., the set of nonnegative integers.

We are interested in determining an online estimate $x_e(t) \in \mathbb{R}^n$, i.e., the estimate at time t has to be a function of the available information (input and output) at the same time instant. This implies that the estimate is generated by means of a device (known as filter) processing the current input and output of the system and generating a state estimate. The filter may be instantaneous, i.e., the estimate is generated instantaneously by processing the available information. In this case we have a static filter. Alternatively, the state estimate can be generated processing the available information through a dynamical device. In this case we have a dynamic filter.

Assume, for simplicity, that $D = 0$. This assumption is without loss of generality. In fact, if $y = Cx + Du$ and u are measurable, then also $\tilde{y} = Cx$ is measurable. Assume, in addition, that the filter which generates the online estimate is linear, finite-dimensional, and time-invariant. Then we may have the following two configurations:

- *Static filter.* The state estimate is generated via the relation

$$x_e = My + Nu, \tag{2}$$

  with M and N constant matrices of appropriate dimensions. The resulting interconnected system is described by the equations

$$\sigma x = Ax + Bu,$$
$$x_e = MCx + Nu. \tag{3}$$

- *Dynamic filter.* The state estimate is generated by the system

$$\sigma\xi = F\xi + Ly + Hu,$$
$$x_e = M\xi + Ny + Pu, \tag{4}$$

with F, L, H, M, N and P constant matrices of appropriate dimensions. The resulting interconnected system is described by the equations

$$\sigma x = Ax + Bu,$$
$$\sigma\xi = F\xi + LCx + Hu, \tag{5}$$
$$x_e = M\xi + NCx + Pu.$$

In what follows we study in detail the dynamic filter configuration. This is mainly due to the fact that this configuration allows us to solve most estimation problems for linear systems. Moreover, while the use of a static filter is very appealing, it provides a useful alternative only in very specific situations.

State Observer

A state observer is a filter that allows to estimate, asymptotically or in finite time, the state of a system from measurements of the input and output signals.

The simplest possible observer can be constructed considering a copy of the system, the state of which has to be estimated. This means that a candidate observer for system (1) is given by

$$\sigma\xi = A\xi + Bu$$
$$x_e = \xi. \tag{6}$$

To assess the properties of this candidate state observer, let $e = x - x_e$ be the estimation error and note that $\sigma e = Ae$. As a result, if $e(0) = 0$, then $e(t) = 0$ for all t and for any input signal u. However, if $e(0) \neq 0$, then, for any input signal u, $e(t)$ is bounded only if the system (1) is stable and converges to zero only if the system (1) is asymptotically stable. If these conditions do not hold, the estimation error is not bounded and system (6) does not qualify as a state observer for system (1). The intrinsic limitation of the observer (6) is that it does not use all the available information, i.e., it does not use the knowledge of

the output signal y. This observer is therefore an open-loop observer.

To exploit the knowledge of y, we modify the observer (6) adding a term which depends upon the available information on the estimation error, which is given by $y_e = Cx_e - y$. This modification yields a candidate state observer described by

$$\sigma\xi = A\xi + Bu + Ly_e,$$
$$x_e = \xi. \tag{7}$$

To assess the properties of this candidate state observer, note that $e = x - x_e$ is such that

$$\sigma e = (A + LC)e. \tag{8}$$

The matrix L (known as output injection gain) can be used to shape the dynamics of the estimation error. In particular, we may select L to assign the characteristic polynomial $p(s)$ of $A + LC$. To this end, note that

$$p(s) = \det(sI - (A + LC)) = \det(sI - (A' + C'L')).$$

Hence, there is a matrix L which arbitrarily assigns the characteristic polynomial of $A + LC$ if and only if the system

$$\sigma\xi = A'\xi + C'v$$

is reachable or, equivalently, if and only if the system (1) is observable.

We summarize the above discussion with two formal statements.

Proposition 1 *Consider system (1) and suppose the system is observable. Let $p(s)$ be a monic polynomial of degree n. Then there is a matrix L such that the characteristic polynomial of $A + LC$ is equal to $p(s)$. Note that for single-output systems, the matrix L assigning the characteristic polynomial of $A + LC$ is unique.*

Proposition 2 *System (1) is observable if and only if it is possible to arbitrarily assign the eigenvalues of $A + LC$.*

Detectability

The main goal of a state observer is to provide an online estimate of the state of a system. This goal may be achieved, as discussed in the previous section, if the system is observable. However, observability is not necessary to achieve this goal: in fact the unobservable modes are not modified by the output injection gain. This implies that there exists a matrix L such that system (8) is asymptotically stable if and only if the unobservable modes of system (1) have negative real part, in the case of continuous-time systems, or have modulo smaller than one, in the case of discrete-time systems. To capture this situation, we introduce a new definition.

Definition 1 (Detectability) System (1) is detectable if its unobservable modes have negative real part, in the case of continuous-time systems, or have modulo smaller than one, in the case of discrete-time systems.

Example 1 (Deadbeat observer) Consider a discrete-time system described by equations of the form

$$x(t+1) = Ax(t) + Bu(t),$$
$$y(t) \;\;= Cx(t),$$

and the problem of designing a state observer, described by the equation (7), such that, for any initial condition $x(0)$ and for any u, $e(k) = 0$, for all $k \geq N$, and for some $N > 0$. A state observer achieving this goal is called a deadbeat state observer. To achieve this goal, it is necessary to select L such that $(A + LC)^N = 0$ or, equivalently, such that the matrix $A + LC$ has all eigenvalues equal to 0. Note that $N \leq n$.

Reduced Order Observer

We have shown that, under the hypotheses of observability or detectability, it is possible to design an asymptotic observer of order n for the system (1). However, this observer is somewhat oversized, i.e., it gives an estimate for the n components of the state vector, without making use of the fact that some of these components can be directly determined from the output function, e.g., if $y = x_1$ there is no need to reconstruct x_1. It makes, therefore, sense to design a *reduced order observer*, i.e., a device that estimates only the part of the state vector which is not directly attainable from the output. To this end consider the system (1) with $D = 0$ and assume that the matrix C has p independent rows. This is the case if rank $C = p$, whereas if rank $C < p$ it is always possible to eliminate redundant rows. Then there exists a matrix Q such that, possibly after reordering the state variables,

$$QC = [I \;\; C_2].$$

Let

$$v = Qy = QCx = x_1 + C_2x_2,$$

in which $x_1(t) \in \mathbb{R}^p$ and $x_2(t) \in \mathbb{R}^{n-p}$ denote the first p and the last $n - p$ components of $x(t)$. Observe that the vector v is measurable.

From the definition of v, we conclude that if v and x_2 are known, then x_1 can be easily computed, i.e., there is no need to construct an observer for x_1.

Define now the new coordinates

$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = Tx = \begin{bmatrix} I & C_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and note that, by construction, $v = Qy = \hat{x}_1$. In the new coordinates, the system, with output v, is described by equations of the form

$$\sigma\hat{x}_1 = \tilde{A}_{11}\hat{x}_1 + \tilde{A}_{12}\hat{x}_2 + \tilde{B}_1u,$$
$$\sigma\hat{x}_2 = \tilde{A}_{21}\hat{x}_1 + \tilde{A}_{22}\hat{x}_2 + \tilde{B}_2u,$$
$$v \;\;= \hat{x}_1.$$

O

To construct an observer for $\hat{x}_2$, consider the system

$$\sigma\xi = F\xi + Hv + Gu,$$

with state ξ, driven by u and v, and with output

$$w = \xi + Lv.$$

The idea is to select the matrices F, H, G, and L in such a way that w be an estimate for $\hat{x}_2$. Let $w - \hat{x}_2$ be the observation error. Then

$$
\begin{aligned}
\sigma w - \sigma\hat{x}_2 = \quad & F\xi + Hv + Gu + L\left[\tilde{A}_{11}\hat{x}_1 + \tilde{A}_{12}\hat{x}_2 + \tilde{B}_1 u\right] - \left[\tilde{A}_{21}\hat{x}_1 + \tilde{A}_{22}\hat{x}_2 + \tilde{B}_2 u\right] \\
= \quad & F\xi + \left(H + L\tilde{A}_{11} - \tilde{A}_{12}\right)\hat{x}_1 + \left[L\tilde{A}_{12} - \tilde{A}_{22}\right]\hat{x}_2 + \left[G + L\tilde{B}_1 - \tilde{B}_2\right]u. \quad (9)
\end{aligned}
$$

To have convergence of the estimation error to zero, regardless of the initial conditions and of the input signal, we must have

$$\sigma(w - \hat{x}_2) = F(w - \hat{x}_2) \qquad (10)$$

and F must have all eigenvalues with negative real part, in the case of continuous-time systems, or with modulo smaller than one, in the case of discrete-time systems. Comparing Eqs. (9) and (10), we obtain that the matrices F, H, G, and L must be such that

$$L\tilde{A}_{12} - \tilde{A}_{22} = -F,$$
$$H + L\tilde{A}_{11} - \tilde{A}_{21} = FL,$$
$$G + L\tilde{B}_1 - \tilde{B}_2 = 0.$$

We now show how the previous equations can be solved and how the stability condition of F can be enforced. Detectability of the system implies that the (reduced system) $\sigma\tilde{\xi} = \tilde{A}_{22}\tilde{\xi}$ with output $\tilde{y} = \tilde{A}_{12}\tilde{\xi}$ is detectable. As a result, there exists a matrix L such that the matrix

$$F = \tilde{A}_{22} - L\tilde{A}_{12}$$

has all eigenvalues with negative real part, in the case of continuous-time systems, or with modulo smaller than one, in the case of discrete-time systems. Then the remaining equations are solved by

$$H = FL - L\tilde{A}_{11} + \tilde{A}_{21},$$
$$G = -L\tilde{B}_1 + \tilde{B}_2.$$

Finally, from $\hat{x}_1 = v$ and the estimate w of $\hat{x}_2$, we build an estimate x_e of the state x inverting the transformation T, i.e.,

$$
\begin{bmatrix} x_{1e} \\ x_{2e} \end{bmatrix} = \begin{bmatrix} I & -C_2 \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}.
$$

Summary and Future Directions

The problem of estimating the state of a linear system from input and output measurements can be solved provided a weak observability condition holds. The problem addressed in this entry is the simplest possible estimation problem: the underlying system is linear and all variables are exactly measured. Observers for nonlinear systems and in the presence of signals *corrupted* by noise can also be designed exploiting some of the basic ingredients, such as the notions of error system and of output injection, discussed in this entry.

Cross-References

▶ Controllability and Observability
▶ Estimation, Survey on
▶ Hybrid Observers
▶ Kalman Filters
▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions
▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Recommended Reading

Classical references on observers for linear systems are given below.

Bibliography

Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston

Brockett RW (1970) Finite dimensional linear systems. Wiley, London

Luenberger DG (1963) Observing the state of a linear system. IEEE Trans Mil Electron 8:74–80

Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London

Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York

Optimal Control and Mechanics

Anthony Bloch
Department of Mathematics, The University of Michigan, Ann Arbor, MI, USA

Abstract

There are very natural close connections between mechanics and optimal control as both involve variational problems. This is a huge subject and we just touch on some interesting connections here. A survey and history may be found in Sussman and Willems (1997). Other aspects may be found in Bloch (2003).

Keywords

Nonholonomic integrator; Sub-Riemannian optimal control; Variational problems

Variational Nonholonomic Systems and Optimal Control

Variational nonholonomic problems (i.e., constrained variational problems) are equivalent to optimal control problems under certain regularity conditions. This issue was investigated in Bloch and Crouch (1994), employing the classical results of Rund (1966) and Bliss (1930), which relate classical constrained variational problems to Hamiltonian flows, although not optimal control problems. We outline the simplest relationship and refer to Bloch (2003) for more details.

Let Q be a smooth manifold and TQ its tangent bundle with coordinates $(q^i, \dot{q}^i)$. Let $L : TQ \to \mathbb{R}$ be a given smooth Lagrangian and let $\Phi : TQ \to \mathbb{R}^{n-m}$ be a given smooth function. We consider the classical Lagrange problem:

$$\min_{q(\cdot)} \int_0^T L(q, \dot{q}) dt \qquad (1)$$

subject to the fixed endpoint conditions $q(0) = 0$, $q(T) = q_T$ and subject to the constraints

$$\Phi(q, \dot{q}) = 0.$$

Consider a modified Lagrangian $\Lambda(q, \dot{q}, \lambda) = L(q, \dot{q}) + \lambda \cdot \Phi(q, \dot{q})$ with Euler–Lagrange equations

$$\frac{d}{dt} \frac{\partial \Lambda}{\partial \dot{q}}(q, \dot{q}, \lambda) - \frac{\partial \Lambda}{\partial q}(q, \dot{q}, \lambda) = 0, \ \ \Phi(q, \dot{q}) = 0.$$
$$(2)$$

We can rewrite this equation in Hamiltonian form and show that the resulting equations are equivalent to the equations of motion given by the maximum principle for a suitable optimal control problem. Set $p = \frac{\partial \Lambda}{\partial \dot{q}}(q, \dot{q}, \lambda)$ and consider this equation together with the constraints $\Phi(q, \dot{q}) = 0$. We can solve these two equations for $(\dot{q}, \lambda)$ under suitable conditions as discussed in Bloch (2003). We obtain the standard Hamiltonian equations with $H(q, p) = p \cdot \phi(q, p) - L(q, \phi(q, p))$.

We now compare this to the optimal control problem

O

$$\min_{u(\cdot)} \int_0^T g(q, u) dt \qquad (3)$$

subject to $q(0) = 0$, $q(T) = q_T$, $\dot{q} = f(q, u)$, where $u \in \mathbb{R}^m$ and f, g are smooth functions.

Then we have the following:

Theorem 1 *The Lagrange problem and optimal control problem generate the same (regular) extremal trajectories, provided that:*

 (i) $\Phi(q, \dot{q}) = 0$ *if and only if there exists a u such that* $\dot{q} = f(q, u)$.
 (ii) $L(q, f(q, u)) = g(q, u)$.

For the proof and more details, see Bloch (2003).

The n-Dimensional Rigid Body

An interesting mechanical example is the n-dimensional rigid body. See Manakov (1976) and Ratiu (1980).

One can introduce a related system which we will call *the symmetric representation of the rigid body*; see Bloch et al. (2002).

By definition, ***the left invariant representation of the symmetric rigid body system*** is given by the first-order equations

$$\dot{Q} = Q\Omega; \qquad \dot{P} = P\Omega \qquad (4)$$

where $Q, P \in \mathrm{SO}(n)$ and where Ω is regarded as a function of Q and P via the equations

$$\Omega := J^{-1}(M) \in \mathfrak{so}(n) \quad \text{and} \quad M := Q^T P - P^T Q.$$

One can check that differentiating M yields the classical form of the n-dimensional rigid body equations. For more on the precise relationship, see Bloch et al. (2002).

Now we can link the symmetric representation of the rigid body equations with the theory of optimal control. This work, developed in Bloch and Crouch (1996) and more generally in Bloch et al. (2002), has been further extended to optimal control problems for the infinitesimal generators of group actions (so-called Clebsch optimal control problems) in Gay-Balmaz and Ratiu (2011) and Bloch et al. (2011, 2013) and even further to

a class of embedded control problems in Bloch et al. (2011, 2013).

Let $T > 0$, $Q_0, Q_T \in \mathrm{SO}(n)$ be given and fixed. Let the rigid body optimal control problem be given by

$$\min_{U \in \mathfrak{so}(n)} \frac{1}{4} \int_0^T \langle U, J(U) \rangle dt \qquad (5)$$

subject to the constraint on U that there be a curve $Q(t) \in \mathrm{SO}(n)$ such that

$$\dot{Q} = QU \qquad Q(0) = Q_0, \qquad Q(T) = Q_T. \qquad (6)$$

Proposition 1 *The rigid body optimal control problem has optimal evolution equations (4) where P is the costate vector given by the maximum principle.*

The optimal controls in this case are given by

$$U = J^{-1}(Q^T P - P^T Q). \qquad (7)$$

Kinematic Sub-Riemannian Optimal Control Problems

Optimal control of underactuated kinematic systems give rise to very interesting mechanical systems.

The problem is referred to as sub-Riemannian in that it gives rise to a geodesic flow with respect to a singular metric (see the work of Strichartz (1983, 1987) and Montgomery (2002) and references therein). This problem has an interesting history in control theory (see Brockett 1973, 1981; Baillieul 1975). See also Bloch et al. (1994) and Sussmann (1996) and further references below.

We consider control systems of the form

$$\dot{x} = \sum_{i=1}^m X_i u_i, \quad x \in M, \quad u \in \Omega \subset \mathbb{R}^m, \quad (8)$$

where Ω contains an open subset that contains the origin, M is a smooth manifold of dimension n, and each of the vector fields in the collection $F := \{X_1, \ldots, X_k\}$ is complete.

We assume that the system satisfies the accessibility rank condition and is thus controllable, since there is no drift term. Then we can pose the optimal control problem

$$\min_{u(\cdot)} \int_0^T \frac{1}{2} \sum_{i=1}^m u_i^2(t) dt \qquad (9)$$

subject to the dynamics (8) and the endpoint conditions $x(0) = x_0$ and $x(T) = x_T$. These problems were studied by Griffiths (1983) from the constrained variational viewpoint and from the optimal control viewpoint by Brockett (1981, 1983). In the sub-Riemannian geodesic problem, abnormal extremals play an important role. See work by Strichartz (1983), Montgomery (1994, 1995), Sussmann (1996), and Agrachev and Sarychev (1996).

Example: Optimal Control and a Particle in a Magnetic Field The control analysis of the Heisenberg model or nonholonomic integrator goes back to Brockett (1981) and Baillieul (1975), while a modern treatment of the relationship with a particle in a magnetic field may be found in Montgomery (1993), for example. A nice treatment of the pure mechanical aspects of a particle in a magnetic field may be found in Marsden and Ratiu (1999).

The Heisenberg optimal control equations are a particular case of planar charged particle motion in a magnetic field. This may be seen by considering the slightly more general problem below.

We now consider the optimal control problem

$$\min \int (u^2 + v^2) dt \qquad (10)$$

subject to the equations

$$\dot{x} = u,$$
$$\dot{y} = v,$$
$$\dot{z} = A_1 u + A_2 v, \qquad (11)$$

where $A_1(x, y)$ and $A_2(x, y)$ are smooth functions of x and y. $A_1 = y$ and $A_2 = -x$ recover the Heisenberg/nonholonomic integrator equations. More generally we get the flow of a particle in a magnetic field – it is not hard to carry out the optimal control analysis to see this. Details are in Bloch (2003).

Cross-References

▶ Discrete Optimal Control
▶ Optimal Control and Pontryagin's Maximum Principle
▶ Optimal Control with State Space Constraints
▶ Singular Trajectories in Optimal Control

Bibliography

Agrachev AA, Sarychev AV (1996) Abnormal sub-Riemannian geodesics: Morse index and rigidity. Ann Inst H Poincaré Anal Non Linéaire 13:635–690

Baillieul J (1975) Some optimization problems in geometric control theory. Ph.D. thesis, Harvard University

Baillieul J (1978) Geometric methods for nonlinear optimal control problems. J Optim Theory Appl 25:519–548

Bliss G (1930) The problem of lagrange in the calculus of variations. Am J Math 52:673–744

Bloch AM (2003) (with Baillieul J, Crouch PE, Marsden JE), Nonholonomic mechanics and control. Interdisciplinary applied mathematics. Springer, New York

Bloch AM, Crouch PE (1994) Reduction of Euler–Lagrange problems for constrained variational problems and relation with optimal control problems. In: Proceedings of the 33rd IEEE conference on decision and control, Lake Buena Vista. IEEE, pp 2584–2590

Bloch AM, Crouch PE (1996) Optimal control and geodesic flows. Syst Control Lett 28(2):65–72

Bloch AM, Crouch PE, Ratiu TS (1994) Sub-Riemannian optimal control problems. Fields Inst Commun AMS 3:35–48

Bloch AM, Crouch P, Marsden JE, Ratiu TS (2002) The symmetric representation of the rigid body equations and their discretization. Nonlinearity 15: 1309–1341

O

Bloch AM, Crouch PE, Nordkivst N, Sanyal AK (2011) Embedded geodesic problems and optimal control for matrix Lie groups. J Geom Mech 3:197–223

Bloch AM, Crouch PE, Nordkivst N (2013) Continuous and discrete embedded optimal control problems and their application to the analysis of Clebsch optimal control problems and mechanical systems. J Geom Mech 5:1–38

Brockett RW (1973) Lie theory and control systems defined on spheres. SIAM J Appl Math 25(2): 213–225

Brockett RW (1981) Control theory and singular Riemannian geometry. In: Hilton PJ, Young GS (eds) New directions in applied mathematics. Springer, New York, pp 11–27

Brockett RW (1983) Nonlinear control theory and differential geometry. In: Proceedings of the international congress of mathematicians, Warsaw, pp 1357–1368

Gay-Balmaz F, Ratiu TS (2011) Clebsch optimal control formulation in mechanics. J Geom Mech 3: 41–79

Griffiths PA (1983) Exterior differential systems. Birkhäuser, Boston

Manakov SV (1976) Note on the integration of Euler's equations of the dynamics of an n-dimensional rigid body. Funct Anal Appl 10:328–329

Marsden JE, Ratiu TS (1999) Introduction to mechanics and symmetry. Texts in applied mathematics, vol 17. Springer, New York. (1st edn. 1994; 2nd edn. 1999)

Montgomery R (1993) Gauge theory of the falling cat. Fields Inst Commun 1:193–218

Montgomery R (1994) Abnormal minimizers. SIAM J Control Optim 32:1605–1620

Montgomery R (1995) A survey of singular curves in sub-Riemannian geometry. J Dyn Control Syst 1: 49–90

Montgomery R (2002) A tour of sub-Riemannian geometries, their geodesics and applications. Mathematical surveys and monographs, vol 91. American Mathematical Society, Providence

Ratiu T (1980) The motion of the free n-dimensional rigid body. Indiana U Math J 29:609–627

Rund H (1966) The Hamiltonian–Jacobi theory in the calculus of variations. Krieger, New York

Strichartz R (1983) Sub-Riemannian geometry. J Diff Geom 24:221–263; see also J Diff Geom 30:595–596 (1989)

Strichartz RS (1987) The Campbell–Baker–Hausdorff–Dynkin formula and solutions of differential equations. J Funct Anal 72:320–345

Sussmann HJ (1996) A cornucopia of four-dimensional abnormal sub-Riemannian minimizers. In: Bellaïche A, Risler J-J (eds) Sub-Riemannian geometry. Progress in mathematics, vol 144. Birkhäuser, Basel, pp 341–364

Sussmann HJ, Willems JC (1997) 300 years of optimal control: from the Brachystochrone to the maximum principle. IEEE Control Syst Mag 17:32–44

Optimal Control and Pontryagin's Maximum Principle

Richard B. Vinter
Imperial College, London, UK

Abstract

Pontryagin's Maximum Principle is a collection of conditions that must be satisfied by solutions of a class of optimization problems involving dynamic constraints called optimal control problems. It unifies many classical necessary conditions from the calculus of variations. This article provides an overview of the Maximum Principle, including free-time and nonsmooth versions. A time-optimal control problem is solved as an example to illustrate its application.

Keywords

Dynamic constraints; Hamiltonian system; Maximum principle; Nonlinear systems; Optimization

Optimal Control

A widely used framework for studying minimization problems, encountered in the optimal selection of flight trajectories and other areas of advanced engineering design and operation involving dynamic constraints, is to view them as special cases of the problem:

$$
(P) \begin{cases}
\text{Minimize } J(x(.), u(.)) : \\
= \int_0^T L(t, x(t), u(t))dt + g(x(0), x(T)) \\
\text{over measurable functions } u(.) : \\
\quad [0, T] \to R^m \text{ and} \\
\quad \text{absolutely continuous functions } x(.) : \\
\quad [0, T] \to R^n \text{ satisfying} \\
\dot{x}(t) = f(t, x(t), u(t)) \text{ a.e.,} \\
u(t) \in \Omega \text{ a.e.,} \\
(x(0), x(T)) \in C,
\end{cases}
$$

the data for which comprise a number $T > 0$, functions $f : [0, T] \times R^n \times R^m \to R^n$, $L : [0, T] \times R^n \times R^m \to R$ and $g : R^n \times R^n \to R$ and sets $C \subset R^n$ and $\Omega \subset R^m$.

It is assumed that set C has the functional inequality and equality constraint set representation

$$C = \{(x_0, x_1) \in R^n : \phi^i(x_0, x_1) \leq 0 \\ \text{for } i = 1, 2, \ldots, k_1 \text{ and} \quad (1) \\ \psi^i(x_0, x_1) = 0 \quad \text{for } i = 1, 2, \ldots, k_2 \},$$

in which $\phi^i : R^n \times R^n \to R$, $i = 1, \ldots, k_1$ and $\psi^i : R^n \times R^n \to R$, $i = 1, \ldots, k_2$ are given functions.

A control function is a measurable function $u(.) : [0, T] \to R^m$ satisfying $u(t) \in \Omega$ a.e. $t \in [0, T]$ A state trajectory $x(.)$ associated with a control function $u(.)$ is a solution to the differential equation $\dot{x}(t) = f(t, x(t), u(t))$. A pair of functions $(x(.), u(.))$ comprising a control function $u(.)$ and an associated state trajectory $x(.)$ satisfying the condition $(x(0), x(T)) \in C$ is a feasible process. A feasible process $(\bar{x}(.), \bar{u}(.))$ which achieves the minimum of $J(x(.), u(.))$ over all feasible processes is called a minimizer.

Frequently, the initial state is fixed, i.e., C takes the form

$C = \{x_0\} \times C_1$ for some $x_0 \in R^n$ and some $C_1 \subset R^n$.

In this case, (P) is a minimization problem over control functions. Allowing freedom in the choice of initial state introduces a flexibility into the formulation which is useful in some applications however.

Optimization problems involving dynamic constraints (such as, but not exclusively, those expressed as controlled differential equations) are known as optimal control problems. Various frameworks are available for studying such problems. (P) is of special importance, since it embraces a wide range of significant dynamic optimization problems which are beyond the reach of traditional variational techniques and, at the same time, it is well suited to the

derivation of general necessary conditions of optimality.

The Maximum Principle

The centerpiece of optimal control theory is a set of conditions that a minimizer $(\bar{x}(.), \bar{u}(.))$ must satisfy, known as Pontryagin's Maximum Principle or, simply, the Maximum Principle. It came to prominence through a 1961 book, which appeared in English translation as Pontryagin LS et al. (1962). It bears the name of L S Pontryagin, because of his role as leader of the research group at the Steklov Institute, Moscow, which achieved this advance. But the first proof is attributed to Boltyanskii. For given $\lambda \geq 0$, define the Hamiltonian function $H_\lambda : [0, T] \times R^n \times R^n \times R^m \to R'$

$$H_\lambda(t, x, p, u) := p^T f(t, x, u) - \lambda L(t, x, u).$$

Theorem 1 (The Maximum Principle) *Let $(\bar{x}(.), \bar{u}(.))$ be a minimizer for (P). Assume that the following hypotheses are satisfied:*

(i) g is continuously differentiable.

(ii) ϕ^i, $i = 1, \ldots, k_1$ and ψ^i, $i = 1, \ldots, k_2$, are continuously differentiable.

(iii) With $\tilde{f}(t, x, u) = (L(t, x, u), f(t, x, u))$, $\tilde{f}(., ., .)$ is continuous, $\tilde{f}(t, ., u)$ is continuously differentiable for each (t, u), and there exist $\epsilon > 0$ and $k(.) \in L^1$ such that

$$|\tilde{f}(t, x, u) - \tilde{f}(t, x', u)| \leq k(t)|x - x'|$$

for all $x, x' \in R^n$ such that $|x - \bar{x}(t)| \leq \epsilon$ and $|x' - \bar{x}(t)| \leq \epsilon$, and $u \in \Omega$, a.e. $t \in [0, T]$

(iv) Ω is a Borel set.

Then, there exist a number λ ($\lambda = 0$ or 1), an absolutely continuous arc $p : [0, T] \to R^n$, numbers $\alpha^i \geq 0$ for $i = 1, \ldots, k_1$ and numbers β^i for $i = 1, \ldots, k_2$ satisfying

$$(p(.), \lambda, \{\alpha^i\}, \{\beta^i\}) \neq (0, 0, \{0, \ldots 0\}, \{0, \ldots 0\})$$

and such that the following conditions are satisfied:

The Adjoint Equation:

$$-\dot{p}(t) = \frac{\partial}{\partial x} f^T(t, \bar{x}(t), \bar{u}(t)) p(t)$$

$$-\lambda \frac{\partial}{\partial x} L^T(t, \bar{x}(t), \bar{u}(t)), \quad a.e.,$$

The Maximization of the Hamiltonian Condition:

$$H_\lambda(t, \bar{x}(t), p(t), \bar{u}(t))$$

$$= \max_{u \in \Omega} H_\lambda(t, \bar{x}(t), p(t), u) \quad a.e.,$$

The Transversality Condition:

$$(p^T(0), -p^T(T)) = \lambda \nabla g(\bar{x}(0), \bar{x}(T))$$

$$+ \sum_{i=1}^{k_1} \alpha^i \nabla \phi^i(\bar{x}(0), \bar{x}(T))$$

$$+ \sum_{i=1}^{k_2} \beta^i \nabla \psi^i(\bar{x}(0), \bar{x}(T))$$

and $\alpha^i = 0$ *for all* $i \in \{1, \ldots, k_1\}$ *such that* $\phi^i(\bar{x}(0), \bar{x}(T)) < 0,$ *in which*

$$\nabla h(x_0, x_1)(\bar{x}_0, \bar{x}_1):$$

$$= \left[\frac{\partial}{\partial x_0} h(\bar{x}_0, \bar{x}_1), \frac{\partial}{\partial x_1} h(\bar{x}_0, \bar{x}_1) \right]. \quad (2)$$

If the functions $L(t, x, u)$ *and* $f(t, x, u)$ *are independent of t, then also*

Constancy of the Hamiltonian for Autonomous Problems:

$$H_\lambda(\bar{x}(t), p(t), \bar{u}(t)) = c \quad a.e.$$

for some constant c.

We allow the cases $k_1 = 0$ (no inequality constraints) and $k_2 = 0$ (no equality endpoint constraints). In the first case, the non-degeneracy condition becomes $(p(.), \lambda, \{\beta^i\}) \neq (0, 0, 0)$ and the summation involving the α^i's is dropped from the transversality condition. The second case, or any combination of the two cases, is treated similarly.

Derivation of the costate equation and boundary conditions. A simple way to derive the differential equations for the $p_i(.)$'s is, first, to construct the Hamiltonian $H_\lambda(t, x, p, u) = p^T f(t, x, u) p^T f(t, x, u) - \lambda L(t, x, u)$ and, second, to use the fact that the ith component $p_i(.)$ of the costate $p(t) = [p_1(t), \ldots, p_n(t)]^T$ satisfies the equation:

$$-\dot{p}_i(t) = \frac{\partial}{\partial x_i} H_\lambda(t, \bar{x}(t), p(t), \bar{u}(t))$$

$$\text{for } i = 1, \ldots, n.$$

The preceding equations are of course merely a component-wise statement of the costate equation above. In many applications the endpoint constraints take the form

$$x_i(0) = \xi_0^i \text{ for } i \in J_0 \text{ and } x_i(0) \in R^n \text{ for } i \notin J_0$$

$$x_i(T) = \xi_1^i \text{ for } i \in J_1 \text{ and } x_i(0) \in R^n \text{ for } i \notin J_1$$

for given index sets $J_0, J_1 \subset \{0, \ldots, n\}$ and n-vectors ξ_0^i for $i \in J_0$ and ξ_1^i for $i \in J_1$, i.e., the endpoints of each state trajectory component are either "fixed" or "free." In such cases the rules for setting up the boundary conditions on the $p_i(.)$'s are

$$p_i(0) \in R^n \text{ for } i \in J_0 \text{ and } p_i(0)$$

$$= \lambda \frac{\partial}{\partial x_{0i}} g(\bar{x}(0), \bar{x}(T)) \text{ for } i \notin J_0$$

$$p_i(T) \in R^n \text{ for } i \in J_1 \text{ and } -p_i(T)$$

$$= \lambda \frac{\partial}{\partial x_{1i}} g(\bar{x}(0), \bar{x}(T)) \text{ for } i \notin J_1,$$

i.e., if $x_i(0)$ (respectively $x_i(T)$) is fixed, then $p_i(0)$ (respectively $p_i(T)$) is free, and if $x_i(0)$ (respectively $x_i(T)$) is free, then $p_i(0)$ (respectively $p_i(T)$) is fixed.

The optimal control problem (P) is a generalization of the following problem in the calculus of variations:

$$\begin{cases} \text{Minimize } \int_0^T L(t, x(t), \dot{x}(t)) dt \\ \text{over absolutely continuous arcs } x(.): \\ \quad [0, T] \to R^n \text{ satisfying} \\ (x(0), x(T)) = (a, b). \end{cases} \quad (3)$$

for given $L : [0, T] \times R^n \times R^n \to R$ and $(a, b) \in R^n \times R^n$. This problem is a special case of (P) in which $f(t, x, u) = u$, $\Omega = R^n$, $k_1 = 0$, $k_2 = 2n$ and

$$\left((\psi^1(x_0, x_1), \ldots, \psi^n(x_0, x_1)), \right.$$
$$\left(\psi^{n+1}(x_0, x_1) \ldots, \psi^{2n}(x_0, x_1) \right)$$
$$= \left(x_0^T - a^T, x_1^T - b^T \right).$$

It is a straightforward exercise to deduce from the Maximum Principle, in this special case, that a minimizer satisfies the classical Euler–Lagrange and Weierstrass conditions and also that the minimizer and associate costate arc satisfy Hamilton's system of equations, under an additional uniform convexity hypothesis on $L(t, x, .)$. Thus, the Maximum Principle unifies many of the classical necessary conditions from the calculus of variations and, furthermore, validates them under reduced hypotheses. But it has far-reaching implications, beyond these conditions, because it allows the presence of pathwise constraints on the velocities, expressed in terms of a controlled differential equation and a control constraint set, which are encountered in engineering design, econometrics, and other areas.

The Hamiltonian System

In favorable circumstances, we are justified in setting the cost multiplier $\lambda = 1$ and, furthermore, the maximization of the Hamiltonian condition permits us, for each t, to express u as a function of x and p:

$$u = u^*(t, x, p).$$

The Maximum Principle now asserts that a minimizing arc $\bar{x}(.)$ is the first component of a pair of absolutely continuous functions $(\bar{x}(.), p(.))$ satisfying *Hamilton's system of equations*:

$$(-\dot{p}^T(t), \dot{\bar{x}}^T(t)) = \nabla_{xp} H_1(t, \bar{x}(t), p(t), u^*$$
$$(t, \bar{x}(t), p(t))) \quad \text{a.e.,} \quad (4)$$

in which $\nabla_{xp} H_1$ denotes the gradient of $H(t, x, p, u)$ w.r.t. the vector $[x^T, p^T]^T$ variable for fixed (t, u), together with the endpoint conditions

$$(\bar{x}(0), \bar{x}(T)) \in C \text{ and } (p^T(0), -p^T(T))$$
$$= \lambda \nabla g(\bar{x}(0), \bar{x}(T))$$
$$+ \sum_{i=1}^{k_1} \alpha^i \nabla \phi^i(\bar{x}(0), \bar{x}(T))$$
$$+ \sum_{i=1}^{k_2} \beta^i \nabla \psi^i(\bar{x}(0), \bar{x}(T)),$$

for some nonnegative numbers $\{\alpha^i\}$ and numbers $\{\beta^i\}$ satisfying

$$\alpha^i = 0 \quad \text{for all } i \in \{1, \ldots, k_1\} \text{ such that}$$
$$\times \phi^i(\bar{x}(0), \bar{x}(T)) < 0,$$

where ∇g, $\nabla \phi$ and $\nabla \psi$ etc., are as defined in (2). The minimizing control satisfies the relation

$$\bar{u}(t) = u^*(t, \bar{x}(t), p(t)).$$

Notice that the first-order vector differential equation (4) is a system of $2n$ scalar, first-order differential equations. Let us suppose that $\bar{k}_1$ inequality endpoint constraints are active at $(\bar{x}(0), \bar{x}(T))$. Then, satisfaction of the active constraints and the transversality condition impose $2n + \bar{k}_1 + k_2$ on the boundary values of $(\bar{x}(.), p(.))$. Taking account of the fact, however, that there are $\bar{k}_1 + k_2$ unknown endpoint multipliers, we see that the effective number of endpoint constraints accompanying the differential equation (4) is

$$2n + \bar{k}_1 + k_2 - (\bar{k}_1 + k_2) = 2n.$$

Thus, the set of $2n$ scalar first-order differential equations (4) defining the "two-point boundary value problem" to determine $(\bar{x}, p)$ has the "right" number of endpoint conditions.

Refinements

Free-Time Problems: Consider a variant on the "autonomous" case of problem (P) (L and f do not depend on t), call it (FT), in which the terminal time T is no longer fixed, but is a choice variable along with the control function and the initial state, and the cost function is

$$\tilde{J}(T, x(.), u(.))$$

$$:= \int_0^T L(x(t), u(t))dt + \tilde{g}(T, x(0), x(T))$$

for some function $\tilde{g}(.,.,.)$. Take a minimizer $(\bar{T}, \bar{x}(.), \bar{u}(.))$ for (FT). Assume, in addition to hypotheses (i)–(iii), that Ω is bounded and the function $k(.)$ in (iii) is bounded. Then the Maximum Principle conditions (for data in which the end time is frozen at $T = \bar{T}$) continue to be satisfied for some $p(.) : [0, \bar{T}] \to R^n$ and λ, including the constancy of the Hamiltonian condition

$$H_\lambda(\bar{x}(t), p(t), \bar{u}(t)) = c \quad \text{a.e } t \in [0, \bar{T}]$$

for some constant c. But a new condition is required to reflect the extra degree of freedom in the new problem specification, namely, the free end time. This is an additional transversality condition involving the constant value c of the Hamiltonian:

Free Time Transversality Condition: $c = \lambda \frac{\partial}{\partial T} g(\bar{T}, \bar{x}(0), \bar{x}(T))$.

Other Refinements: Versions of the Maximum Principle are available to take account of pathwise functional inequality constraints on state variables ("pure" state constraints) and of both state and control variables ("mixed" constraints). Maximum Principle-like conditions have also been derived for optimal control problems in which the dynamic constraint takes the form of a retarded differential equation with control terms and in which the class of control functions is enlarged to include Dirac delta functions ("impulse" optimal control problems).

The Nonsmooth Maximum Principle

In early derivations of the Maximum Principle, it was assumed that the functions $f(t, x, u)$ and $L(t, x, u)$ were continuously differentiable with respect to the x variable. A major research endeavor since the early 1970s has been to find versions of the Maximum Principle than remain valid when the functions $f(t, x, u)$ and $L(t, x, u)$ satisfy merely a "bounded slope" or, synonymously, a Lipschitz continuity condition with respect to x. Such functions are "nonsmooth" in the sense that they can fail to be differentiable, in the conventional sense, at some points in their domains. An overview of the Maximum Principle would be incomplete without reference to such advances.

The search for nonsmooth optimality conditions is motivated by a desire to solve optimal control problems where, in particular, the function $f(t, x, u)$ is a piecewise linear function of x (for fixed (t, u)). Such functions arise, for example, when the $f(t, x, u)$ is constructed empirically via a lookup table and linear interpolation. Nonsmooth cost integrands are encountered when they are constructed using "pointwise" supremum and/or "absolute value" operations. The function

$$J(x(.)) = \int_0^T |x(t)|dt + \max\{x(1), 0\},$$

which penalizes the L^1 norm of the state trajectory and the terminal value of the scalar state, but only if this is nonnegative, is a case in point.

When attempting to generalize the Maximum Principle to allow for nonsmooth data, we encounter the challenge of interpreting the adjoint equation, which can be written as

$$-\dot{p}(t) = \frac{\partial}{\partial x} H_\lambda(t, \bar{x}(t), \bar{u}(t) p(t)),$$

in circumstances when the x-gradients of f and L are not defined, at least not in a conventional sense. One approach to dealing with this problem is via the Clarke generalized gradient ∂m of function $m: R^n \to R$ at a point $\bar{x}$:

$\partial m(\bar{x}) := \text{co} \{\xi \mid$ there exist sequences $x^i \to \bar{x}$, $\xi^i \to \xi$ such that, for each i, $m(.)$ is Frêchet differentiable at x^i and $\xi_i = \frac{\partial}{\partial x} m(x^i)\}$.

Here, "co" means closed convex hull. In a landmark paper, Clarke FH 1976, Clarke proved a necessary condition commonly referred to as the nonsmooth Maximum Principle, in which the adjoint equation is replaced by a differential inclusion involving the (partial) generalized gradient $\partial_x H(t, \bar{x}(t), p(t), \bar{u}(t))$ of $H(t, ., p(t), \bar{u}(t))$ w.r.t x, evaluated at $\bar{x}(t)$, namely,

$$-\dot{p}^T(t) \in \partial_x H(t, \bar{x}(t), \bar{u}(t)) \quad \text{a.e. } t \in [0, T].$$

This formulation of the "adjoint inclusion" for the nonsmooth Maximum Principle and the unrestricted hypothesis under which it is derived in this paper remain state of the art.

Example

We illustrate the application of the Maximum Principle with a simple example. It has the following interpretation. A 1 kg mass is located 1 m along the line and has zero velocity. We seek a time $\bar{T} > 0$ s. which is the minimum over all times $T > 0$ having the property: there exists a time-varying force $u(t)$, $0 \le t \le 1$ satisfying

$$-1 \le u(t) \le +1$$

such that, under the action of the force, the mass is located at the origin with zero velocity at time T. Note that, in consequence of Newton's second law, the vector $x(t) = (x_1(t), x_2(t))$ comprising the displacement and velocity of mass satisfies

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t). \tag{5}$$

This is a special case of the free-time problem

$$\begin{cases} \text{Minimize } T \\ \text{over times } T > 0, \text{ measurable functions } u(.): \\ \qquad [0, T] \to R \text{ and} \\ \qquad \text{absolutely continuous functions } x(.): \\ \qquad [0, T] \to R^2 \text{ such that} \\ \dot{x}(t) = Ax(t) + bu(t) \quad \text{a.e.} \\ u(t) \in \Omega \quad \text{a.e.} \\ (x_1(0), x_2(0)) = (1, 0) \quad \text{and} \\ \qquad (x_1(T), x_2(T)) = (0, 0). \end{cases}$$

in which $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $\Omega = [-1, +1]$.

The (free-time) Maximum Principle provides the following information about a minimizing end time $\bar{T}$, control $\bar{u}(.)$, and corresponding state $\bar{x}(.) = (\bar{x}_1(.), \bar{x}_2(.))$. There exists an arc $p(.) = [p_1(.), p_2(.)]^T$ such that

$$\dot{\bar{x}}_1(t) = \bar{x}_2(t) \quad \text{and} \quad \dot{\bar{x}}_2(t) = \bar{u}(t), \tag{6}$$

$$-\dot{p}_1(t) = 0 \quad \text{and} \quad -\dot{p}_2(t) = p_1(t), \tag{7}$$

$$\bar{u}(t) = \arg\max \{p_2(t)u \mid u \in [-1, +1]\} \tag{8}$$

$$(\bar{x}_1, \bar{x}_2)(0) = (1, 0) \text{ and } (\bar{x}_1, \bar{x}_2)(T) = (0, 0) \tag{9}$$

$$p_1(t)\bar{x}_2(t) + |p_2(t)| = \lambda \quad \text{for all } t. \tag{10}$$

Condition (1) permits us to express $\bar{u}(.)$ in terms of $p_2(.)$, thus

$$\bar{u}(t) = \text{sign}\{p_2(t)\},$$

and thereby eliminate $\bar{u}(.)$. It can be shown that relations (6)–(2) have a unique solution for $\bar{T}$, $\bar{u}(t)$, $\bar{x}(t)$, $p(t)$ and $\lambda = 0$ or 1. Furthermore, these relations cannot be satisfied with $\lambda = 0$. The unique solution (with $\lambda = 1$) is

$$\bar{T} = 2$$

$$(\bar{x}_1(t), \bar{x}_2(t))$$

$$= \begin{cases} (1 - \frac{1}{2}t^2, -t) & \text{if } t \in [0, 1) \\ (\frac{1}{2} - (t-1) + \frac{1}{2}(t-1)^2, \\ \quad -1 + \frac{1}{2}(t-1)) & \text{if } t \in [1, 2], \end{cases}$$

$$\bar{u}(t) = \begin{cases} -1 \text{ if } t \in [0, 1) \\ +1 \text{ if } t \in [1, 2], \end{cases}$$

$$p_1(t) = -1 \text{ and } p_2(t) = -1 + t \quad \text{for } t \in [0, 2].$$

The Maximum Principle is a necessary condition of optimality. Since a minimizer exists and since $(\bar{T}, \bar{x}(.), \bar{u}(.), p(.))$ is a unique solution to the Maximum Principle relations, it follows that $(\bar{T}, \bar{x}(.), \bar{u}(.))$ is the solution to the problem.

This problem is amenable to simpler, more elementary, solution techniques. But the above solution is enlightening, because it highlights important generic features of the Maximum Principle. We see how the "maximization of the Hamiltonian condition" can be used to eliminate the control function and thereby to set up a two-point boundary problem for $\bar{x}(.)$ and $p(.)$ (a very nonclassical construction).

Cross-References

Bibliography

Clarke FH (1976) The maximum principle under minimal hypotheses. SIAM J Control Optim 14:1078–1091
Pontryagin VG et al. (1962) The Mathematical Theory of Optimal Processes, K. N. Tririgoff, Transl., L. W. Neustadt, Ed., Wiley, New York, 1962
Reflections on the origins of the Maximum Principle appear in:
Pesch HJ, Plail M (2009) The maximum principle of optimal control: a history of ingenious ideas and missed opportunities. Control Cybern 38:973–995
Expository texts on the Maximum Principle and related control theory include:
Berkovitz LD (1974) Optimal control theory. Applied mathematical sciences, vol 12. Springer, New York
Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York
Ioffe AD, Tihomirov VM (1979) Theory of extremal problems. North-Holland, Amsterdam
Ross IM (2009) A primer on Pontryagins principle in optimal control. Collegiate Publishers, San Francisco
For expository texts that also cover advances in the theory of necessary conditions related to the Maximum

Principle, based on techniques of Nonsmooth Analysis we refer to:
Clarke FH (1983) Optimization and nonsmooth analysis. Wiley-nterscience, New York
Clarke FH (2013) Functional analysis, calculus of variations and optimal control. Graduate texts in mathematics. Springer, London
Vinter RB (2000) Optimal control. Birkhaüser, Boston
Engineering texts illustrating the application of the Maximum Principle to solve problems of optimal control and design, in flight mechanics and other areas, include:
Bryson AE, Ho Y-C (1975) Applied optimal control (Revised edn). Halstead Press (a division of John Wiley and Sons), New York
Bryson AE (1999) Dynamic optimization. Addison Wesley Longman, Menlo Park

---

Optimal Control and the Dynamic Programming Principle

Maurizio Falcone
Dipartimento di Matematica, SAPIENZA – Università di Roma, Rome, Italy

Abstract

This entry illustrates the application of Bellman's dynamic programming principle within the context of optimal control problems for continuous-time dynamical systems. The approach leads to a characterization of the optimal value of the cost functional, over all possible trajectories given the initial conditions, in terms of a partial differential equation called the Hamilton–Jacobi–Bellman equation. Importantly, this can be used to synthesize the corresponding optimal control input as a state-feedback law.

Keywords

Continuous-time dynamics; Hamilton–Jacobi–Bellman equation; Optimization; Nonlinear systems; State feedback

Introduction

The dynamic programming principle (DPP) is a fundamental tool in optimal control theory. It was largely developed by Richard Bellman

in the 1950s (Bellman 1957) and has since been applied to various problems in deterministic and stochastic optimal control. The goal of optimal control is to determine the control function and the corresponding trajectory of a dynamical system which together optimize a given criterion usually expressed in terms of an integral along the trajectory (the cost functional) (Fleming and Rishel 1975; Macki and Strauss 1982). The function which associates with the initial condition of the dynamical system the optimal value of the cost functional among all the possible trajectories is called the *value function*. The most interesting point is that via the dynamic programming principle, one can derive a characterization of the value function in terms of a nonlinear partial differential equation (the Hamilton–Jacobi–Bellman equation) and then use it to synthesize a feedback control law. This is the major advantage over the approach based on the Pontryagin Maximum Principle (PMP) (Boltyanskii et al. 1956; Pontryagin et al. 1962). In fact, the PMP merely gives necessary conditions for the characterization of the open-loop optimal control and of the corresponding optimal trajectory. The DPP has also been applied to construct approximation schemes for the value function although this approach suffers from the "curse of dimensionality" since one has to solve a nonlinear partial differential equation in a high dimension. Despite the elegance of the DPP approach, its practical application is limited by this bottleneck, and the solution of many optimal control problems has been accomplished instead via the two-point boundary value problem associated with the PMP.

The Infinite Horizon Problem

Let us present the main ideas for the classical *infinite horizon problem*. Let a controlled dynamical system be given by

$$\begin{cases} \dot{y}(s) = f(y(s), \alpha(s)) \\ y(t_0) = x_0. \end{cases} \quad (1)$$

where $x_0, y(s) \in \mathbb{R}^d$, and

$$\alpha : [t_0, T] \to A \subseteq \mathbb{R}^m,$$

with T finite or $+\infty$. Under the assumption that the control is measurable, existence and uniqueness properties for the solution of (1) are ensured by the Carathèodory theorem:

Theorem 1 (Carathèodory) *Assume that:*
1. *$f(\cdot, \cdot)$ is continuous.*
2. *There exists a positive constant $L_f > 0$ such that*

$$|f(x, a) - f(y, a)| \leq L_f |x - y|,$$

   for all $x, y \in \mathbb{R}^d$, $t \in \mathbb{R}^+$ and $a \in A$.
3. *$f(x, \alpha(t))$ is measurable with respect to t.*

   Then, there is a unique absolutely continuous function $y : [t_0, T] \to \mathbb{R}^d$ that satisfies

$$y(s) = x_0 + \int_{t_0}^{s} f(y(\tau), \alpha(\tau)) d\tau. \quad (2)$$

which is interpreted as the solution of (1).

Note that the solution is continuous, but only a.e. differentiable, so it must be regarded as a weak solution of (1). By the theorem above, fixing a control in the set of admissible controls

$$\alpha \in \mathcal{A} := \{\alpha : [t_0, T] \to A, \text{measurable}\}$$

yields a unique trajectory of (1) which is denoted by $y_{x_0, t_0}(s; \alpha)$. Changing the control policy generates a family of solutions of the controlled system (1) with index α. Since the dynamics (1) are "autonomous," the initial time t_0 can be shifted to 0 by a change of variable. So to simplify the notation for autonomous dynamics, we can set $t_0 = 0$ and we denote this family by $y_{x_0}(s; \alpha)$ (or even write it as $y(s)$ if no ambiguity over the initial state or control arises). It is customary in dynamic programming, moreover, to use the notations x and t instead of x_0 and t_0 (since x and t appear as variables in the Hamilton–Jacobi–Bellman equation).

Optimal control problems require the introduction of a *cost functional* $J : \mathcal{A} \to \mathbb{R}$ which is used to select the "optimal trajectory" for (1). In the case of the infinite horizon problem, we set $t_0 = 0$, $x_0 = x$, and this functional is defined as

O

$$J_x(\alpha) = \int_0^\infty g(y_x(s, \alpha), \alpha(s))e^{-\lambda s}\,ds \quad (3)$$

for a given $\lambda > 0$. The function g represents the *running cost* and λ is the *discount factor*, which can be used to take into account the reduced value, at the initial time, of future costs. From a technical point of view, the presence of the discount factor ensures that the integral is finite whenever g is bounded. Note that one can also consider the undiscounted problem ($\lambda = 0$) provided the integral is still finite. The goal of optimal control is to find an optimal pair (y^*, α^*) that minimizes the cost functional. If we seek optimal controls in open-loop form, i.e., as functions of t, then the Pontryagin Maximum Principle furnishes necessary conditions for a pair (y^*, α^*) to be optimal.

A major drawback of an open-loop control is that being constructed as a function of time, it cannot take into account errors in the true state of the system, due, for example, to model errors or external disturbances, which may take the evolution far from the optimal forecasted trajectory. Another limitation of this approach is that a new computation of the control is required whenever the initial state is changed.

For these reasons, we are interested in the so-called *feedback controls*, that is, controls expressed as functions of the state of the system. Under feedback control, if the system trajectory is perturbed, the system reacts by changing its control strategy according to the change in the state. One of the main motivations for using the DPP is that it yields solutions to optimal control problems in the form of feedback controls.

DPP for the Infinite Horizon Problem

The starting point of dynamic programming is to introduce an auxiliary function, the *value function*, which for our problem is

$$v(x) = \inf_{\alpha \in \mathcal{A}} J_x(\alpha), \quad (4)$$

where, as above, x is the initial position of the system. The value function has a clear meaning: it is the optimal cost associated with the initial

position x. This is a reference value which can be useful to evaluate the efficiency of a control – if $J_x(\bar{\alpha})$ is close to $v(x)$, this means that $\bar{\alpha}$ is "efficient."

Bellman's dynamic programming principle provides a first characterization of the value function.

Proposition 1 (DPP for the infinite horizon problem) *Under the assumptions of Theorem 1, for all $x \in \mathbb{R}^d$ and $\tau > 0$,*

$$v(x) = \inf_{\alpha \in \mathcal{A}} \left\{ \int_0^\tau g(y_x(s; \alpha), \alpha(s))e^{-\lambda s}\,ds \right.$$
$$\left. + e^{-\lambda \tau} v(y_x(\tau; \alpha)) \right\}. \quad (5)$$

Proof Denote by $\bar{v}(x)$ the right-hand side of (5). First, we remark that for any $x \in \mathbb{R}^d$ and $\bar{\alpha} \in \mathcal{A}$,

$$J_x(\bar{\alpha}) = \int_0^\infty g(\bar{y}(s), \bar{\alpha}(s))e^{-\lambda s}\,ds$$
$$= \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s))e^{-\lambda s}\,ds$$
$$+ \int_\tau^\infty g(\bar{y}(s), \bar{\alpha}(s))e^{-\lambda s}\,ds$$
$$= \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s))e^{-\lambda s}\,ds + e^{-\lambda \tau}$$
$$\times \int_0^\infty g(\bar{y}(s + \tau), \bar{\alpha}(s + \tau))e^{-\lambda s}\,ds$$
$$\geq \int_0^\tau g(\bar{y}(s), \bar{\alpha}(s))e^{-\lambda s}\,ds + e^{-\lambda \tau} v(\bar{y}(\tau))$$

(here, $y_x(s, \bar{\alpha})$ is abbreviated as $\bar{y}(s)$). Taking the infimum over all trajectories, first over the right-hand side and then the left of this inequality, yields

$$v(x) \geq \bar{v}(x) \quad (6)$$

To prove the opposite inequality, we recall that $\bar{v}$ is defined as an infimum, and so, for any $x \in \mathbb{R}^d$ and $\varepsilon > 0$, there exists a control $\bar{\alpha}_\varepsilon$ (and the corresponding evolution $\bar{y}_\varepsilon$) such that

$$\bar{v}(x) + \varepsilon \geq \int_0^\tau g(\bar{y}_\varepsilon(s), \bar{\alpha}_\varepsilon(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(\bar{y}_\varepsilon(\tau)). \tag{7}$$

On the other hand, the value function v being also defined as an infimum, for any $x \in \mathbb{R}^d$ and $\varepsilon > 0$, there exists a control $\tilde{\alpha}_\varepsilon$ such that

$$v(\bar{y}_\varepsilon(\tau)) + \varepsilon \geq J_{\bar{y}_\varepsilon(\tau)}(\tilde{\alpha}_\varepsilon). \tag{8}$$

Inserting (8) in (7), we get

$$\begin{aligned} \bar{v}(x) &\geq \int_0^\tau g(\bar{y}_\varepsilon(s), \bar{\alpha}_\varepsilon(s)) e^{-\lambda s} ds \\ &\quad + e^{-\lambda \tau} J_{\bar{y}_\varepsilon(\tau)}(\tilde{\alpha}_\varepsilon) - (1 + e^{-\lambda \tau}) \varepsilon \\ &\geq J_x(\hat{\alpha}) - (1 + e^{-\lambda \tau}) \varepsilon \\ &\geq v(x) - (1 + e^{-\lambda \tau}) \varepsilon, \end{aligned} \tag{9}$$

where $\hat{\alpha}$ is a control defined by

$$\hat{\alpha}(s) = \begin{cases} \bar{\alpha}_\varepsilon(s) & 0 \leq s < \tau \\ \tilde{\alpha}_\varepsilon(s - \tau) & s \geq \tau. \end{cases} \tag{10}$$

(Note that $\hat{\alpha}(\cdot)$ is still measurable). Since ε is arbitrary, (9) finally yields $\bar{v}(x) \geq v(x)$.

We observe that this proof crucially relies on the fact that the control defined by (10) still belongs to $\mathcal{A}$, being a measurable control. The possibility of obtaining an admissible control by joining together two different measurable controls is known as the *concatenation property*.

The Hamilton–Jacobi–Bellman Equation

The DPP can be used to characterize the value function in terms of a nonlinear partial differential equation. In fact, let $\alpha^* \in \mathcal{A}$ be the optimal control, and y^* the associated evolution (to simplify, we are assuming that the infimum is a minimum). Then,

$$v(x) = \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds + e^{-\lambda \tau} v(y^*(\tau)),$$

that is,

$$v(x) - e^{-\lambda \tau} v(y^*(\tau)) = \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds$$

so that adding and subtracting $e^{-\lambda \tau} v(x)$ and dividing by τ, we get

$$\begin{aligned} e^{-\lambda \tau} \frac{(v(x) - v(y^*(\tau)))}{\tau} &+ \frac{v(x)(1 - e^{-\lambda \tau})}{\tau} \\ &= \frac{1}{\tau} \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds. \end{aligned}$$

Assume now that v is regular. By passing to the limit as $\tau \to 0^+$, we have

$$\lim_{\tau \to 0^+} -\frac{v(y^*(\tau)) - v(x)}{\tau}$$
$$= -Dv(x) \cdot \dot{y}^*(x) = -Dv(x) \cdot f(x, \alpha^*(0))$$

$$\lim_{\tau \to 0^+} v(x) \frac{(1 - e^{-\lambda \tau})}{\tau} = \lambda v(x)$$

$$\lim_{\tau \to 0^+} \frac{1}{\tau} \int_0^\tau g(y^*(s), \alpha^*(s)) e^{-\lambda s} ds = g(x, \alpha^*(0))$$

where we have assumed that $\alpha^*(\cdot)$ is continuous at 0. Then, we can conclude

$$\lambda v(x) - Dv(x) \cdot f(x, a^*) - g(x, a^*) = 0 \tag{11}$$

where $a^* = \alpha^*(0)$. Similarly, using the equivalent form

$$v(x) + \sup_{\alpha \in \mathcal{A}} \left\{ -\int_0^\tau g(y(s), \alpha(s)) e^{-\lambda s} ds \right.$$
$$\left. - e^{-\lambda \tau} v(y(\tau)) \right\} = 0$$

of the DPP and the inequality, this implies for any (continuous at 0) control $\alpha \in \mathcal{A}$,

$$\begin{aligned} \lambda v(x) &- Dv(x) \cdot f(x, a) - g(x, a) \\ &\leq 0, \quad \text{for every } a \in A. \end{aligned} \tag{12}$$

Combining (11) and (12), we obtain the *Hamilton–Jacobi–Bellman equation* (or *dynamic programming equation*):

$$\lambda u(x) + \sup_{a \in A} \{-f(x,a) \cdot Du(x) - g(x,a)\} = 0,$$
$$(13)$$

which characterizes the value function for the infinite horizon problem associated with minimizing (3). Note that given x, the value of a achieving the max (assuming it exists) corresponds to the control $a^* = \alpha^*(0)$, and this makes it natural to interpret the argmax in (13) as the optimal feedback at x (see Bardi and Capuzzo Dolcetta (1997) for more details).

In short, (13) can be written as

$$H(x,u,Du) = 0$$

with $x \in \mathbb{R}^d$, and

$$H(x,u,p) = \lambda u(x) + \sup_{a \in A} \{-f(x,a) \cdot p - g(x,a)\}.$$
$$(14)$$

Note that $H(x,u,\cdot)$ is convex (being the sup of a family of linear functions) and that $H(x,\cdot,p)$ is monotone (since $\lambda > 0$). It is also easy to see that the solution u is not differentiable even when f and g are smooth functions (i.e., $f, g, \in C^\infty(\mathbb{R}^n, A)$), so we need to deal with weak solution of the Bellman equation. This can be done in the framework of viscosity solutions, a theory initiated by Crandall and Lions in the 1980s which has been successfully applied in many areas as optimal control, fluid dynamics, and image processing (see the books Barles (1994) and Bardi and Capuzzo Dolcetta (1997) for an extended introduction and numerous applications to optimal control). Typically viscosity solutions are Lipschitz continuous solutions so they are differentiable almost everywhere.

An Extension to the Minimum Time Problem

In the minimum time problem, we want to minimize the time of arrival of the state on a given *target set* $\mathcal{T}$. We will assume that $\mathcal{T} \subset \mathbb{R}^d$ is a closed set. Then our cost functional will be given by

$$J(x,\alpha) = t_x(\alpha)$$

where

$$t_x(\alpha) := \begin{cases} \min\{t : y_x(t,\alpha) \in \mathcal{T}\} & \text{if } y_x(t,\alpha) \in \mathcal{T} \\ & \text{for some } t \geq 0 \\ +\infty & \text{if } y_x(t,\alpha) \notin \mathcal{T} \\ & \text{for any } t \geq 0 \end{cases}$$

The corresponding value function is called the *minimum time function*

$$T(x) := \inf_{\alpha(\cdot) \in \mathcal{A}} t_x(\alpha(\cdot)). \qquad (15)$$

The main difference with respect to the previous problem is that now the value function T will be finite valued only on a subset $\mathcal{R}$ which depends on the target, on the dynamics, and on the set of admissible controls.

Definition 1 The reachable set $\mathcal{R}$ is defined by

$$\mathcal{R} := \cup_{t>0} \mathcal{R}(t) = \{x \in \mathbb{R}^n : T(x) < +\infty\}$$

where, for $t > 0$, $\mathcal{R}(t) := \{x \in \mathbb{R}^n : T(x) < t\}$.

The meaning is clear: $\mathcal{R}$ is the set of initial points which can be driven to the target in finite time. The system is said to be *controllable* on $\mathcal{T}$ if for all $t > 0$, $\mathcal{T} \subset \text{int}(\mathcal{R}(t))$ (here, $\text{int}(D)$ denotes the interior of the set D). Assuming controllability in a neighborhood of the target one gets the continuity of the minimum time function and under the assumptions made on f, A, and $\mathcal{T}$, one can prove some interesting properties:

  (i) $\mathcal{R}$ is open.
 (ii) T is continuous on $\mathcal{R}$.
(iii) $\lim_{x \to x_0} T(x) = +\infty$, for any $x_0 \in \partial\mathcal{R}$.

Now let us denote by $\mathcal{X}_D$ the characteristic function of the set D. Using in $\mathcal{R}$ arguments similar to the proof of DPP in the previous section one can obtain the following DPP:

Proposition 2 (DPP for the minimum time problem) *For any $x \in \mathcal{R}$, the value function satisfies*

$$T(x) = \inf_{\alpha \in \mathcal{A}} \{t \wedge t_x(\alpha) + \mathcal{X}_{\{t \leq t_x(\alpha)\}} T(y_x(t,\alpha))\}$$
$$\text{for any } t \geq 0 \qquad (16)$$

and

$$T(x) = \inf_{\alpha \in \mathcal{A}} \{t + T(y_x(t, \alpha))\}$$

$$\text{for any } t \in [0, T(x)] \qquad (17)$$

From the previous DPP, one can also obtain the following characterization of the minimum time function.

Proposition 3 *Let* $\mathcal{R} \backslash \mathcal{T}$ *be open and* $T \in C(\mathcal{R} \backslash \mathcal{T})$*, then* T *is a viscosity solution of*

$$\max_{a \in A} \{-f(x, a) \cdot \nabla T(x)\} = 1 \qquad x \in \mathcal{R} \backslash \mathcal{T}$$

$$(18)$$

coupled with the natural boundary condition

$$\begin{cases} T(x) = 0 & x \in \partial \mathcal{T} \\ \lim_{x \to \partial \mathcal{R}} T(x) = +\infty \end{cases}$$

By the change of variable $v(x) = 1 - e^{-T(x)}$, one can obtain a simpler problem getting rid of the boundary condition on $\partial \mathcal{R}$ (which is unknown). The new function v will be the unique viscosity solution of an external Dirichlet problem (see Bardi and Capuzzo Dolcetta (1997) for more details), and the reachable set can be recovered a posteriori via the relation $\mathcal{R} = \{x \in \mathbb{R}^d : v(x) < 1\}$.

Further Extensions and Related Topics

The DPP has been extended from deterministic control problems to many other problems. In the framework of stochastic control problems where the dynamics are given by a diffusion, the characterization of the value function obtained via the DPP leads to a second-order Hamilton–Jacobi–Bellman equation (Fleming and Soner 1993; Kushner and Dupuis 2001). Another interesting extension has been made in differential games where the DPP is based on the delicate notion of nonanticipative strategies for the players and leads to a nonconvex nonlinear partial differential equation (the Isaacs equation)

(Bardi and Capuzzo Dolcetta 1997). For a short introduction to numerical methods based on DP and exploiting the so-called "value iteration," we refer the interested reader to the Appendix A in Bardi and Capuzzo Dolcetta (1997) and to Kushner and Dupuis (2001) (see also the book Howard (1960) for the "policy iteration").

Cross-References

▶ Numerical Methods for Nonlinear Optimal Control Problems

▶ Optimal Control and Pontryagin's Maximum Principle

Bibliography

Bardi M, Capuzzo Dolcetta I (1997) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Birkhäuser, Boston

Barles G (1994) Solutions de viscosité des équations de Hamilton-Jacobi. In: Mathématiques et applications, vol 17. Springer, Paris

Bellman R (1957) Dynamic programming. Princeton University Press, Princeton

Bertsekas DP (1987) Dynamic programming: deterministic and stochastic models. Prentice Hall, Englewood Cliffs

Boltyanskii VG, Gamkrelidze RV, Pontryagin LS (1956) On the theory of optimal processes (in Russian). Doklady Akademii Nauk SSSR 110, 7–10

Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York

Fleming WH, Soner HM (1993) Controlled Markov processes and viscosity solutions. Springer, New York

Howard RA (1960) Dynamic programming and Markov processes. Wiley, New York

Kushner HJ, Dupuis P (2001) Numerical methods for stochastic control problems in continuous time. Springer, Berlin

Macki J, Strauss A (1982) Introduction to optimal control theory. Springer, Berlin/Heidelberg/New York

Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1961) Matematicheskaya teoriya optimal' nykh prozessov. Fizmatgiz, Moscow. Translated into English. The mathematical theory of optimal processes. John Wiley and Sons (Interscience Publishers), New York, 1962

Ross IM (2009) A primer on Pontryagin's principle in optimal control. Collegiate Publishers, San Francisco

Optimal Control via Factorization and Model Matching

Michael Cantoni
Department of Electrical & Electronic
Engineering, The University of Melbourne,
Parkville, VIC, Australia

Abstract

One approach to linear control system design involves the matching of certain input-output models with respect to a quantification of closed-loop performance. The approach is based on a parametrization of all stabilizing feedback controllers, which relies on the existence of coprime factorizations of the plant model. This parametrization and spectral factorization methods for solving model-matching problems are described within the context of impulse-response energy and worst-case energy-gain measures of controller performance.

Keywords

Coprime factorization; $\mathscr{H}_2$ control; $\mathscr{H}_\infty$ control; Spectral factorization; Youla-Kučera controller parametrization

Introduction

Various linear control problems can be formulated in terms of the interconnection shown in Fig. 1; e.g., see Francis and Doyle (1987), Boyd and Barratt (1991), and Zhou et al. (1996). The linear system K is a *controller* (with input y and output $u - v_1$) to be designed for the generalized *plant* model G. The latter is constructed so that controller *performance* (i.e., the quality of K relative to specifications) can be quantified as a nonnegative functional of

$$H(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}, \tag{1}$$

which relates the input w and the output z when $v_1 = 0$ and $v_2 = 0$. The objective is to select K, to minimize this measure of performance. Alternatively, controllers that achieve a specified upper bound are sought. It is also usual to require *internal stability*, which pertains to the fictitious signals v_1 and v_2, as discussed more subsequently. The best known examples are $\mathscr{H}_2$ and $\mathscr{H}_\infty$ control problems. In the former, performance is quantified as the energy (resp. power) of z when w is impulsive (resp. unit white noise), and in the latter, as the worst-case energy gain from w to z, which can be used to reflect robustness to model uncertainty; see Zhou et al. (1996).

The special case of $G_{22} = 0$ gives rise to a (weighted) *model-matching* problem, in that the corresponding performance map $H(G, K) = G_{11} + G_{12}KG_{21}$ exhibits *affine* dependence on the design variable K, which is chosen to match $G_{12}KG_{21}$ to $-G_{11}$ with respect to the scalar quantification of performance. Any internally stabilizable problem with $G_{22} \neq 0$, can be converted into a model-matching problem. The key ingredients in this transformation are coprime factorizations of the plant model. The role of these and other factorizations in a model-matching approach to $\mathscr{H}_2$ and $\mathscr{H}_\infty$ control problems is the focus of this article.

For the sake of argument, finite-dimensional linear time-invariant systems are considered via real-rational transfer functions in the *frequency domain*, as the existence of all factorizations
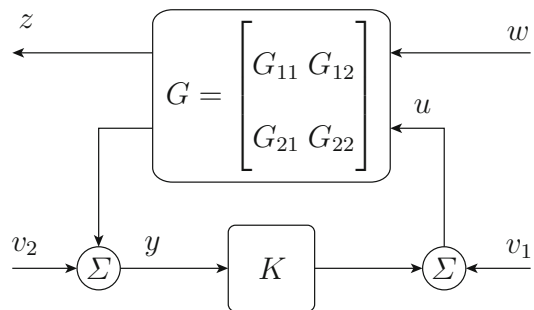
Optimal Control via Factorization and Model Matching, Fig. 1 Standard interconnection for control system design

employed is well understood in this setting. Indeed, constructions via state-space realizations and Riccati equations are well known. The merits of the model-matching approach pursued here are at least twofold: (i) the underlying algebraic input-output perspective extends to more abstract settings, including classes of distributed-parameter and time-varying systems (Desoer et al. 1980; Vidyasagar 1985; Curtain and Zwart 1995; Feintuch 1998; Quadrat 2006); and (ii) model matching is a convex problem for various measures of performance (including mixed indexes) and controller constraints. The latter can be exploited to devise numerical algorithms for controller optimization (Boyd and Barratt 1991; Dahleh and Diaz-Bobillo 1995; Qi et al. 2004).

First, some notation regarding transfer functions and two measures of performance for control system design is defined. Coprime factorizations are then described within the context of a well-known parametrization of stabilizing controllers, originally discovered by Youla et al. (1976) and Kucera (1975). This yields an affine parametrization of performance maps for problems in standard form, and thus, a transformation to a model-matching problem. Finally, the role of spectral factorizations in solving model-matching problems with respect to impulse-response energy ($\mathscr{H}_2$) and worst-case energy-gain ($\mathscr{H}_\infty$) measures of performance is discussed.

Notation and Nomenclature

$\mathcal{R}$ generically denotes a linear space of matrices having fixed row and column dimensions, which are not reflected in the notation for convenience, and entries that are *proper* real-rational functions of the complex variable s; i.e., $\left(\sum_{k=1}^m b_k s^k\right) / \left(\sum_{k=1}^n a_k s^k\right)$ for sets of real coefficients $\{a_k\}_{k=1}^n$ and $\{b_k\}_{k=1}^m$ with $m \leq n < \infty$. The compatibility of matrix dimensions is implicitly assumed henceforth. All matrices in $\mathcal{R}$ have (nonunique) "state-space" realizations of the form $C(sI - A)^{-1}B + D$, where A, B, C and D are real valued matrices.

This form naturally arises in frequency-domain analysis of the *input-output* map associated with the time-domain model $\dot{x}(t) = Ax(t) + Bu(t)$, with initial condition $x(0) = 0$ and output equation $y(t) = Cx(t) + Du(t)$, where $\dot{x}$ denotes the time derivative of x and u is the input. The study of such linear time-invariant differential equation models via the Laplace transform and *multiplication* by real-rational transfer function matrices is fundamental in linear systems theory (Kailath 1980; Francis 1987; Zhou et al. 1996). $P \in \mathcal{R}$ has an inverse $P^{-1} \in \mathcal{R}$ if and only if $\lim_{|s| \to \infty} P(s)$ is a nonsingular matrix. The superscripts T and $*$ denote the transpose and complex conjugate transpose. For a matrix $Z = Z^*$ with complex entries, $Z > 0$ means $z^* Z z \geq \epsilon z^* z$ for some $\epsilon > 0$ and all complex vectors z of compatible dimension. $P^\sim(s) := P(-s)^T$, whereby $(P(j\omega))^* = P^\sim(j\omega)$ for all real ω with $j := \sqrt{-1}$. Zeros of transfer function denominators are called poles.

In subsequent sections, several subspaces of $\mathcal{R}$ are used to define and solve two standard linear control problems. The subspace $\mathcal{B} \subset \mathcal{R}$ comprises transfer functions that have no poles on the imaginary axis in the complex plane. For $P \in \mathcal{B}$, the scalar performance index

$$\|P\|_\infty := \max_{-\infty \leq \omega \leq \infty} \bar{\sigma}(P(j\omega)) \geq 0$$

is finite; the real number $\bar{\sigma}(Z)$ is the maximum singular value of the matrix argument Z. This index measures the worst-case energy-gain from an input signal u, to the output signal $y = Pu$. Note that $\|P\|_\infty < \gamma$ if and only if $\gamma^2 I - P^\sim(j\omega)P(j\omega) > 0$ for all $-\infty \leq \omega \leq \infty$.

The subspace $\mathcal{S} \subset \mathcal{B} \subset \mathcal{R}$ consists of transfer functions that have no poles with positive real part. A transfer function in $\mathcal{S}$ is called *stable* because the corresponding input-output map is causal in the time domain, as well as bounded-in-bounded-out (in various senses). If $P \in \mathcal{S}$ is such that $P^\sim P = I$, then it is called *inner*. If $P, P^{-1} \in \mathcal{S}$, then both are called *outer*.

Let $\mathcal{L}$ denote the subspace of *strictly-proper* transfer functions in $\mathcal{B}$; i.e., for all entries of the

matrix, the degree n of the denominator *exceeds* the degree m of the numerator. Observe that $P \in \mathcal{L}$ if and only if $P^\sim \in \mathcal{L}$. Moreover, $P_1 P_2 \in \mathcal{L}$ and $P_3 P_1 \in \mathcal{L}$ for all $P_1 \in \mathcal{L}$ and $P_i \in \mathcal{B}$, $i = 2, 3$. Now, for $P_1, P_2 \in \mathcal{L}$, define the inner-product

$$\langle P_1, P_2 \rangle := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}(P_1^\sim(j\omega) P_2(j\omega)) d\omega < \infty$$

and the scalar performance index $\|P\|_2 := \sqrt{\langle P, P \rangle} \geq 0$ for $P \in \mathcal{L}$. This index equates to the root-mean-square (energy) measure of the impulse response and the covariance (power) of the output signal $y = Pu$, when the input signal u is unit white noise. By the properties $\text{trace}(Z_1 + Z_2) = \text{trace}(Z_1) + \text{trace}(Z_2)$ and $\text{trace}(Z_1 Z_2) = \text{trace}(Z_2 Z_1)$ of the matrix trace, it follows that $\langle P_1 + P_2, P_3 \rangle = \langle P_1, P_3 \rangle + \langle P_2, P_3 \rangle$ and

$$\langle P_1, P_2 P_3 \rangle = \langle P_2^\sim P_1, P_3 \rangle = \langle P_1 P_3^\sim, P_2 \rangle$$
$$= \langle P_3^\sim, P_1^\sim P_2 \rangle \text{ for } P_i \in \mathcal{L}, \; i = 1, 2, 3. \tag{2}$$

The (not closed) subspace $\mathcal{L} \subset \mathcal{B} \subset \mathcal{R}$ can be expressed as the direct sum $\mathcal{L} = \mathcal{H} + \mathcal{H}_\perp$, where $\mathcal{H} = \mathcal{L} \cap \mathcal{S}$ and $\mathcal{H}_\perp$ is the subspace of transfer functions in $\mathcal{L}$ that have no poles with negative real part. That is, given $P \in \mathcal{L}$, there is a unique decomposition $P = \boldsymbol{\Pi}_+(P) + \boldsymbol{\Pi}_-(P)$, with $\boldsymbol{\Pi}_+(P) \in \mathcal{H}$ and $\boldsymbol{\Pi}_-(P) \in \mathcal{H}_\perp$. Observe that $P \in \mathcal{H}$ if and only if $P^\sim \in \mathcal{H}_\perp$. It can be shown via Plancherel's theorem that $\langle P_1, P_2 \rangle = 0$ for $P_1 \in \mathcal{H}_\perp$ and $P_2 \in \mathcal{H}$. Finally, note that $P_1 P_2 \in \mathcal{H}$ and $P_3 P_1 \in \mathcal{H}$ for $P_1 \in \mathcal{H}$ and $P_i \in \mathcal{S}$, $i = 2, 3$.

Coprime and Spectral Factorizations

Given $P \in \mathcal{R}$, the factorizations $P = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ are said to be (doubly) *coprime* over $\mathcal{S}$, if $N, M, \tilde{N}, \tilde{M}$ are all elements of $\mathcal{S}$ and there exist $U_0, V_0, \tilde{U}_0, \tilde{V}_0$ all in $\mathcal{S}$ such that

$$\begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \end{bmatrix} \begin{bmatrix} M \\ N \end{bmatrix} = I \text{ and } \begin{bmatrix} -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = I \tag{3}$$

hold; i.e., $\begin{bmatrix} M^T & N^T \end{bmatrix}$ and $\begin{bmatrix} -\tilde{N} & \tilde{M} \end{bmatrix}$ are right invertible in $\mathcal{S}$. Importantly, if the factorizations are coprime and $P \in \mathcal{S}$, then $M^{-1} = \tilde{V}_0 - \tilde{U}_0 P$ and $\tilde{M}^{-1} = V_0 - P U_0$ are in $\mathcal{S}$, as sums of products of transfer functions in $\mathcal{S}$; i.e., M and $\tilde{M}$ are outer. Doubly coprime factorizations over $\mathcal{S}$ always exist, but these are not unique. Constructions from state-space realizations can be found in Zhou et al. (1996, Chapter 6) and Francis (1987), for example. As mentioned above, coprime factorizations play a role in transforming a standard problem into the special case of a model matching problem, via the Youla-Kučera parametrization of internally stabilizing controllers presented in the next section.

Subsequently, a special coprime factorization proves to be useful. If $P^\sim(s)P(s) = M^{-\sim}(s)N^\sim(s)N(s)M^{-1}(s) > 0$ for s on the extended imaginary axis (i.e., for $s = j\omega$ with $-\infty \leq \omega \leq \infty$), then it is possible to choose the factor N to be inner. In this case, if P is also an element of $\mathcal{S}$, then $P = NM^{-1}$ is called an *inner-outer factorization*, and $P^\sim P = (M^{-1})^\sim M^{-1}$ is called a *spectral factorization*, since $M, M^{-1} \in \mathcal{S}$. More generally, if $\Xi = \Xi^\sim \in \mathcal{B}$ satisfies $\Xi(s) > 0$ for s on the extended imaginary axis, then there exists a (non-unique) spectral factor $\Sigma, \Sigma^{-1} \in \mathcal{S}$ such that $\Xi = \Sigma^\sim \Sigma$. Similarly, there exists a co-spectral factor $\tilde{\Sigma}, \tilde{\Sigma}^{-1} \in \mathcal{S}$ such that $\Xi = \tilde{\Sigma} \tilde{\Sigma}^\sim$. State-space constructions via Riccati equations can be found in Zhou et al. (1996, Chapter 13), for example.

Affine Controller/Performance-Map Parametrization

With reference to Fig. 1, a generalized plant model $G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \in \mathcal{R}$ is said to be *internally stabilizable* if there exists a $K \in \mathcal{R}$ such that the nine transfer functions associated with the map from the vector of signals (w, v_1, v_2) to

the vector of signals (z, u, y), which includes the performance map $H(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}$, are all elements of $\mathcal{S}$. Accounting in this way for the influence of the fictitious signals v_1 and v_2, and the behavior of the internal signals u and y, amounts to following requirement: Given minimal state-space realizations, any nonzero initial condition response decays exponentially in the time domain when G and K are interconnected according to Fig. 1 with $w = 0$, $v_1 = 0$ and $v_2 = 0$. Not every $G \in \mathcal{R}$ is internally stabilizable in the sense just defined; for example, take G_{11} to have a pole with positive real part and $G_{21} = G_{12} = G_{22} = 0$. A necessary condition for stabilizability is $(I - G_{22}K)^{-1} \in \mathcal{R}$; i.e., the inverse must be proper. The latter always holds if G_{22} is strictly proper, as assumed henceforth to simplify the presentation. It is also assumed that G is internally stabilizable.

It can be shown that G is internally stabilized by K if and only if the standard feedback interconnection of G_{22} and K, corresponding to $w = 0$ in Fig. 1, is internally stable. That is, if and only if the transfer function

$$\begin{bmatrix} I & -K \\ -G_{22} & I \end{bmatrix} \in \mathcal{R}, \qquad (4)$$

which relates u and y to v_1 and v_2 by virtue of the summing junctions at the interconnection points, has an inverse in $\mathcal{S}$; see Francis (1987, Theorem 4.2). Substituting the coprime factorizations $K = UV^{-1} = \tilde{V}^{-1}\tilde{U}$ and $G_{22} = NM^{-1} = \tilde{M}^{-1}\tilde{N}$, it follows that the inverse of (4) is an element of $\mathcal{S}$ if and only if

$$\begin{bmatrix} M & U \\ N & V \end{bmatrix}^{-1} \in \mathcal{S} \qquad \Leftrightarrow \qquad \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \in \mathcal{S}. \tag{5}$$

The equivalent characterizations of internal stability in (5) lead directly to affine parametrizations of controllers and performance maps. Specifically, following the approach of Desoer et al. (1980), Vidyasagar (1985), and Francis (1987), suppose that the factorizations $G_{22} = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ are *doubly coprime* in the

sense that (3) holds for some $U_0, V_0, \tilde{U}_0, \tilde{V}_0 \in \mathcal{S}$. Indeed, since $0 = G_{22} - G_{22} = \tilde{M}^{-1}(\tilde{M}N - \tilde{N}M)M^{-1}$, it follows that

$$\begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U_0 \\ N & V_0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} M & U_0 \\ N & V_0 \end{bmatrix} \begin{bmatrix} \tilde{V}_0 & -\tilde{U}_0 \\ -\tilde{N} & \tilde{M} \end{bmatrix}. \tag{6}$$

Exploiting this and the condition (5), it holds that $K = UV^{-1}$ stabilizes G_{22} if and only if

$$U = (U_0 - MQ) \text{ and } V = (V_0 - NQ) \text{ with } Q \in \mathcal{S}.$$

Similarly, K stabilizes G_{22} if and only if $K = (\tilde{V}_0 - Q\tilde{N})^{-1}(\tilde{U}_0 - Q\tilde{M})$ with $Q \in \mathcal{S}$. Together, these constitute the Youla-Kučera parametrizations of internally stabilizing controllers. Importantly, the coprime factors that appear in these are affine functions of the stable parameter Q. Moreover, using (6), an affine parametrization of the standard performance map (1) holds by direct substitution of either controller parametrization. Specifically,

$$H(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}$$
$$= T_1 + T_2QT_3 \quad \text{with } Q \in \mathcal{S}, \quad (7)$$

where $T_1 = G_{11} + G_{12}U_0\tilde{M}G_{21}$, $T_2 = -G_{12}M$ and $T_3 = \tilde{M}G_{21}$. Clearly, $T_1 \in \mathcal{S}$ since this is the performance map when $Q = 0 \in \mathcal{S}$. By the assumption that G is stabilizable, it follows that T_2 and T_3 are also elements of $\mathcal{S}$; see Francis (1987, Chapter 4). The so-called Q-parametrization in (7) motivates the subsequent consideration of model-matching problems with respect to the standard measures of control system performance $\|\cdot\|_2$ and $\|\cdot\|_\infty$.

Model-Matching via Spectral Factorization

Bearing in mind the Q-parametrization (7), consider the following $\mathscr{H}_2$ model-matching problem,

where inf denotes greatest lower bound (infimum) and $T_i \in \mathcal{S}, i = 1, 2, 3$:

$$\inf_{Q \in \mathcal{S}} \|T_1 + T_2 Q T_3\|_2.$$

Assume that $T_2(s)$ and $T_3(s)$ have full column and row rank, respectively, for s on the extended imaginary axis. Also assume that T_1 is strictly proper, whereby Q must be strictly proper, and thus an element of $\mathcal{H} \subset \mathcal{S}$, for the performance index to be finite. Under this standard collec-

tion of assumptions, the infimum is achieved as shown below.

A minimizer of the convex functional $f := Q \in \mathcal{H} \mapsto \langle (T_1 + T_2 Q T_3), (T_1 + T_2 Q T_3) \rangle$ is a solution of the model matching problem. Given spectral factorizations $\Phi^\sim \Phi = T_2^\sim T_2$ and $\Lambda \Lambda^\sim = T_3 T_3^\sim$ (i.e., $\Phi, \Phi^{-1}, \Lambda, \Lambda^{-1} \in \mathcal{S}$), which exist by the assumptions on the problem data, let $R := \Phi Q \Lambda$ and $W := \Phi^{-\sim} T_2^\sim T_1 T_3^\sim \Lambda^{-\sim}$. Then for $Q \in \mathcal{H}$, which is equivalent to $R \in \mathcal{H}$ by the properties of spectral factors, it follows that

$$
\begin{aligned}
f(Q) &= \langle T_1, T_1 \rangle + \langle \Phi^{-\sim} T_2^\sim T_1 T_3^\sim \Lambda^{-\sim}, R \rangle + \langle R, \Phi^{-\sim} T_2^\sim T_1 T_3^\sim \Lambda^{-\sim} \rangle + \langle R, R \rangle \qquad (8) \\
&= \langle T_1, T_1 \rangle + \langle (\boldsymbol{\Pi}_-(W) + \boldsymbol{\Pi}_+(W) + R), (\boldsymbol{\Pi}_-(W) + \boldsymbol{\Pi}_+(W) + R) \rangle - \langle W, W \rangle \\
&= \langle T_1, T_1 \rangle - \langle \boldsymbol{\Pi}_+(W), \boldsymbol{\Pi}_+(W) \rangle + \langle (\boldsymbol{\Pi}_+(W) + R), (\boldsymbol{\Pi}_+(W) + R) \rangle, \qquad (9)
\end{aligned}
$$

where the second last equality holds by "completion-of-squares" and the last equality holds since $\langle \boldsymbol{\Pi}_+(W), \boldsymbol{\Pi}_-(W) \rangle = 0 = \langle R, \boldsymbol{\Pi}_-(W) \rangle$. From (9) it is apparent that

$$Q = -\Phi^{-1} \boldsymbol{\Pi}_+(\Phi^{-\sim} T_2^\sim T_1 T_3^\sim \Lambda^{-\sim}) \Lambda^{-1}$$

is a minimizer of f. As above, spectral factorization is a key component of the so-called Wiener-Hopf approach of Youla et al. (1976) and DeSantis et al. (1978).

Now consider the $\mathcal{H}_\infty$ model-matching problem

$$\inf_{Q \in \mathcal{S}} \|T_1 + T_2 Q T_3\|_\infty,$$

given $T_i \in \mathcal{S}, i = 1, 2, 3$. This is more challenging than the problem discussed above, where $\| \cdot \|_2$ is the performance index. While sufficient conditions are again available for the infimum to be achieved, computing a minimizer is generally difficult; see Francis and Doyle (1987) and Glover et al. (1991). As such, nearly optimal solutions are often sought by considering the relaxed problem of finding the set of $Q \in \mathcal{S}$ that satisfy $\|T_1 + T_2 Q T_3\|_\infty < \gamma$ for a value of $\gamma > 0$ greater than, but close to, the infimum.

With a view to highlighting the role of factorization methods and simplifying the presentation, suppose that T_2 is inner, which is possible without loss of generality via inner-outer factorization if $T_2(s)$ has full column rank for s on the extended imaginary axis. Furthermore, assume that $T_3 = I$. Following the approach of Francis (1987) and Green et al. (1990), let $X^\sim = \begin{bmatrix} X_1^\sim & X_2^\sim \end{bmatrix} := \begin{bmatrix} T_2 & I - T_2 T_2^\sim \end{bmatrix} \in \mathcal{B}$, so that $X^\sim X = I$ and $X T_2 = \begin{bmatrix} I \\ 0 \end{bmatrix}$. Observe that

$$
\begin{aligned}
\|T_1 + T_2 Q\|_\infty &= \|X(T_1 + T_2 Q)\|_\infty \\
&= \left\| \begin{bmatrix} T_2^\sim T_1 + Q \\ (I - T_2 T_2^\sim) T_1 \end{bmatrix} \right\|_\infty < \gamma
\end{aligned}
\qquad (10)
$$

if and only if

$$
\begin{aligned}
0 < \gamma^2 I &- T_1^\sim (I - T_2 T_2^\sim) T_1 \\
&- (T_2^\sim T_1 + Q)^\sim (T_2^\sim T_1 + Q) \qquad (11)
\end{aligned}
$$

on the extended imaginary axis. Note that (11) implies $0 < \gamma^2 I - T_1^\sim (I - T_2 T_2^\sim)^2 T_1$. Thus, it follows that there exists a $Q \in \mathcal{S}$ for which (10) holds if and only if the following are both satisfied: (a) there exists a spectral factorization

$\gamma^2 \Psi^\sim \Psi = \gamma^2 I - T_1^\sim (I - T_2 T_2^\sim)^2 T_1$; and (b) there exists an $\bar{R}(= Q\Psi^{-1}) \in \mathcal{S}$ such that $\|\bar{W} + \bar{R}\|_\infty < \gamma$, where $\bar{W} := T_2^\sim T_1 \Psi^{-1} \in \mathcal{B}$. The condition (b) is a well-known extension problem and a solution exists if and only if the induced norm of the Hankel operator with symbol $\bar{W}$ is less than γ, which is part of a result known as Nehari's theorem. In fact, (b) is equivalent to the existence of a spectral factor $\Upsilon, \Upsilon^{-1} \in \mathcal{S}$ with $\Upsilon_{11}^{-1} \in \mathcal{S}$ such that

$$\Upsilon^\sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \Upsilon = \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}^\sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}, \tag{12}$$

in which case $\|\bar{W} + \bar{R}\|_\infty \leq \gamma$ if and only if $\bar{R} = \bar{R}_1 \bar{R}_2^{-1}$ with $\begin{bmatrix} \bar{R}_1^T & \bar{R}_2^T \end{bmatrix} := \begin{bmatrix} \bar{S}^T & I \end{bmatrix} \Upsilon^{-T}$, $\bar{S} \in \mathcal{S}$ and $\|\bar{S}\|_\infty \leq \gamma$; see Ball and Ran (1987), Francis (1987), and Green et al. (1990) for details, including state-space constructions of the factors via Riccati equations. Noting that

$$\begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix}^\sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix}^\sim \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix}^\sim$$
$$\begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} I & \bar{W} \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix},$$

it follows using (12) that there exists a $Q \in \mathcal{S}$ such that (10) holds if and only if there exists a spectral factor $\Omega, \Omega^{-1} \in \mathcal{S}$ with $\Omega_{11}^{-1} \in \mathcal{S}$ $\left(\Omega = \Upsilon \begin{bmatrix} I & 0 \\ 0 & \Psi \end{bmatrix}\right)$ that satisfies

$$\begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix}^\sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} T_2 & T_1 \\ 0 & I \end{bmatrix} = \Omega^\sim \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \Omega, \tag{13}$$

in which case $\|T_1 + T_2 Q\|_\infty \leq \gamma$ if and only if $Q = Q_1 Q_2^{-1}$, where $\begin{bmatrix} Q_1^T & Q_2^T \end{bmatrix} := \begin{bmatrix} S^T & I \end{bmatrix} \Omega^{-T}$, $S \in \mathcal{S}$ and $\|S\|_\infty \leq \gamma$; see Green et al. (1990). So-called J-spectral factorizations of the kind in (12) and (13) also appear in the chain-scattering/conjugation approach of Kimura (1989, 1997) and the factorization approach of Ball et al. (1991), for example.

Summary

The preceding sections highlight the role of co-prime and spectral factorizations in formulating and solving model-matching problems that arise from standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. The transformation of standard control problems to model-matching problems hinges on an affine parametrization of internally stabilized performance maps. Beyond the problems considered here, this parametrization can be exploited to devise numerical algorithms for various other control problems in terms of convex mathematical programs.

Cross-References

- ▶ H-Infinity Control
- ▶ H$_2$ Optimal Control
- ▶ Polynomial/Algebraic Design Methods
- ▶ Spectral Factorization

Bibliography

Ball JA, Ran ACM (1987) Optimal Hankel norm model reductions and Wiener-Hopf factorization I: the canonical case. SIAM J Control Optim 25(2):362–382

Ball JA, Helton JW, Verma M (1991) A factorization principle for stabilization of linear control systems. Int J Robust Nonlinear Control 1(4):229–294

Boyd SP, Barratt CH (1991) Linear controller design: limits of performance. Prentice Hall, Englewood Cliffs

Curtain RF, Zwart HJ (1995) An introduction to infinite-dimensional linear systems theory. Volume 21 of texts in applied mathematics. Springer, New York

Dahleh MA, Diaz-Bobillo IJ (1995) Control of uncertain systems: a linear programming approach. Prentice Hall, Upper Saddle River

DeSantis RM, Saeks R, Tung LJ (1978) Basic optimal estimation and control problems in Hilbert space. Math Syst Theory 12(1):175–203

Desoer C, Liu R-W, Murray J, Saeks R (1980) Feedback system design: the fractional representation approach to analysis and synthesis. IEEE Trans Autom Control 25(3):399–412

Feintuch A (1998) Robust control theory in Hilbert space. Applied mathematical sciences. Spinger, New York

Francis BA (1987) A course in H_∞ control theory. Lecture notes in control and information sciences. Springer, Berlin/New York

O

Francis BA, Doyle JC (1987) Linear control theory with an H_∞ optimality criterion. SIAM J Control Optim 25(4):815–844

Glover K, Limebeer DJN, Doyle JC, Kasenally EM, Safonov MG (1991) A characterization of all solutions to the four block general distance problem. SIAM J Control Optim 29(2): 283–324

Green M, Glover K, Limebeer DJN, Doyle JC (1990) A J-spectral factorization approach to $\mathscr{H}_\infty$ control. SIAM J Control Optim 28(6):1350–1371

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Kimura H (1989) Conjugation, interpolation and model-matching in H_∞. Int J Control 49(1): 269–307

Kimura H (1997) Chain-scattering approach to H^∞ control. Systems & control. Birkhäuser, Boston

Kucera V (1975) Stability of discrete linear control systems. In: 6th IFAC world congress, Boston. Paper 44.1

Qi X, Salapaka MV, Voulgaris PG, Khammash M (2004) Structured optimal and robust control with multiple criteria: a convex solution. IEEE Trans Autom Control 49(10):1623–1640

Quadrat A (2006) On a generalization of the Youla–Kučera parametrization. Part II: the lattice approach to MIMO systems. Math Control Signals Syst 18(3):199–235

Vidyasagar M (1985) Control system synthesis: a factorization approach. Signal processing, optimization and control. MIT, Cambridge

Youla D, Jabr H, Bongiorno J (1976) Modern Wiener-Hopf design of optimal controllers – Part II: the multivariable case. IEEE Trans Autom Control 21(3):319–338

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Optimal Control with State Space Constraints

Heinz Schättler
Washington University, St. Louis, MO, USA

Abstract

Necessary and sufficient conditions for optimality in optimal control problems with state space constraints are reviewed with emphasis on geometric aspects.

Keywords

Problem Formulation and Terminology

Many practical problems in engineering or of scientific interest can be formulated in the framework of optimal control problems with state space constraints. Examples range from the space shuttle reentry problem in aeronautics (Bonnard et al. 2003) to the problem of minimizing the base transit time in bipolar transistors in electronics (Rinaldi and Schättler 2003).

An optimal control problem with state space constraints in Bolza form takes the following form: minimize a functional

$$J(u) = \int_{t_0}^{T} L(t, x(t), u(t)) dt + \Phi(T, x(T))$$

over all Lebesgue measurable functions $u : [t_0, T] \to U$ that take values in a control set $U \subset \mathbb{R}^m$, subject to the dynamics

$$\dot{x}(t) = F(t, x(t), u(t)), \qquad x(t_0) = x_0,$$

terminal constraints

$$\Psi(T, x(T)) = 0,$$

and state space constraints

$$h_\alpha(t, x(t)) \leq 0 \qquad \text{for } \alpha = 1, \ldots, r.$$

The focus of this contribution is on state space constraints, and, for simplicity, in this formulation, we have omitted mixed control state space constraints of the form $g_\beta(t, x, u) \leq 0$. States x lie in $\mathbb{R}^n$ and controls in $\mathbb{R}^m$; typically, the control set $U \subset \mathbb{R}^m$ is compact and convex, often a polyhedron. The time-varying vector field $F : \mathbb{R} \times \mathbb{R}^n \times U \to \mathbb{R}^n$ is continuously differentiable in (t, x), and the terminal constraint $N = \{(t, x) : \Psi(t, x) = 0\}$ is defined by continuously differentiable mappings $\psi_i : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^k$ with

the property that the gradients $\nabla \psi_i = (\frac{\partial \psi_i}{\partial t}, \frac{\partial \psi_i}{\partial x})$ (which we write as row vectors) are linearly independent on N. The terminal time T can be free or fixed; a fixed terminal time simply would be prescribed by one of the functions ψ_i. The state space constraints

$$M_\alpha = \{(t, x) : h_\alpha(t, x) = 0\}, \qquad \alpha = 1, \dots, r,$$

are defined by continuously differentiable time-varying vector fields $h_\alpha : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$, $(t, x) \mapsto h_\alpha(t, x)$, and we assume that the gradients ∇h_α do not vanish on M_α. In particular, each set M_α thus is an embedded submanifold of codimension 1 of $\mathbb{R}^{n+1}$. We denote by $h = (h_1, \dots, h_r)^T$ the time-varying vector field defining the state space constraints.

Terminology: *Admissible controls* are locally bounded Lebesgue measurable functions that take values in the control set, $u : [t_0, T] \to U$. Given any admissible control, the initial value problem $\dot{x}(t) = F(t, x(t), u(t))$, $x(t_0) = x_0$, has a unique solution defined on some maximal open interval of definition I. This solution is called the *trajectory* corresponding to the control u and the pair (x, u) is a *controlled trajectory*. An arc Γ of the graph of a trajectory defined over an open interval I for which none of the state space constraints is active is called an *interior arc*, and Γ is a *boundary arc* if at least one constraint is active on all of I. We call Γ an M_α-boundary arc over I if only the constraint $h_\alpha \leq 0$ is active on I. The times τ when interior arcs and boundary arcs meet are called *junction times* and the corresponding pairs $(\tau, x(\tau))$ *junction points*.

Despite the abundance and importance of practical problems that can be described as optimal control problems with state space constraints, for such problems the theory still lacks the coherence that the theory for problems without state space constraints has reached and there still exist significant gaps between the theories of necessary and sufficient conditions for optimality for optimal control problems with state space constraints. The theory of existence of optimal solutions differs little between optimal control problems with and without state space constraints, is well established, and will not be addressed here (e.g., see Cesari 1983 or the Filippov-Cesari theorem in Hartl et al. 1995).

Necessary Conditions for Optimality

First-order necessary conditions for optimality are given by the *Pontryagin maximum principle* (Pontryagin et al. 1962). The zero set of even a smooth (C^∞) function can be an arbitrary closed subset of the state space. As a result, in necessary conditions for optimality, the multipliers associated with the state space constraints a priori are only known to be nonnegative Radon measures (Ioffe and Tikhomirov 1979; Vinter 2000). Let $u_* : [t_0, T] \to U$ be an optimal control with corresponding trajectory x_* and, for simplicity of presentation, also assume that no state constraints are active at the terminal time so that the standard transversality conditions apply. Then it follows that there exist a constant $\lambda_0 \geq 0$, an absolutely continuous function η, which we write as row-vector, $\eta : [t_0, T] \to (\mathbb{R}^n)^*$, and nonnegative Radon measures $\mu_\alpha \in C^*([t_0, T]; \mathbb{R})$, $\alpha = 1, \dots, r$, with support in the sets $R_\alpha = \{t \in [t_0, T] : h_\alpha(t, x_*(t)) = 0\}$, which do not all vanish simultaneously, i.e.,

$$\lambda_0 + \|\eta\|_\infty + \sum_{\alpha=1}^r \mu_\alpha([t_0, T]) > 0,$$

such that with

$$\lambda(t) = \eta(t) - \sum_{\alpha=1}^r \int_{[t_0, t)} \frac{\partial h_\alpha}{\partial x}(s, x_*(s)) d\mu_\alpha(s),$$

and

$$H = H(t, \lambda_0, \lambda, x, u) = \lambda_0 L(t, x, u) + \lambda F(t, x, u)$$

the following conditions hold:

(a) The adjoint equation holds in the form

$$\dot{\eta}(t) = -\frac{\partial H}{\partial x}(t, \lambda_0, \lambda(t), x_*(t), u_*(t))$$

$$= -\lambda_0 \frac{\partial L}{\partial x}(t, x_*(t), u_*(t))$$

$$- \lambda(t) \frac{\partial F}{\partial x}(t, x_*(t), u_*(t)),$$

and there exists a row-vector $\mu \in (\mathbb{R}^k)^*$ such that

$$\lambda(T) = \lambda_0 \frac{\partial \Phi}{\partial x}(T, x_*(T)) + \mu \frac{\partial \Psi}{\partial x}(T, x_*(T))$$

and

$$0 = H(T, \lambda_0, \lambda(T), x_*(T), u_*(T))$$

$$+ \lambda_0 \frac{\partial \Phi}{\partial t}(T, x_*(T)) + \mu \frac{\partial \Psi}{\partial t}(T, x_*(T)).$$

(b) The optimal control minimizes the Hamiltonian over the control set U along $(\lambda(t), x_*(t))$:

$$H(t, \lambda_0, \lambda(t), x_*(t), u_*(t))$$

$$= \min_{v \in U} H(t, \lambda_0, \lambda(t), x_*(t), v).$$

Furthermore,

$$H(t, \lambda_0, \lambda(t), x_*(t), u_*(t))$$

$$= H(T, \lambda_0, \lambda(t), x_*(t), u_*(t))$$

$$- \int_{[t,T]} \frac{\partial H}{\partial t}(s, \lambda_0, \lambda(s), x_*(s), u_*(s)) ds$$

$$+ \sum_{\alpha=1}^{r} \int_{[t,T]} \frac{\partial h_\alpha}{\partial t}(s, x_*(s)) d\mu_\alpha(s)$$

Controlled trajectories (x, u) for which there exist multipliers such that these conditions are satisfied are called *extremals*. In general, it cannot be excluded that λ_0 vanishes and extremals with $\lambda_0 = 0$ are called *abnormal*, while those with $\lambda_0 > 0$ are called *normal*. In this case, the multiplier can be normalized, $\lambda_0 = 1$.

Special Case: A Mayer Problem for Single-Input Control Linear Systems

Under the general assumptions formulated above, the sets $R_\alpha \subset [t_0, T]$ when a particular constraint is active can be arbitrarily complicated.

But in many practical applications, state constraints have strong geometric properties – often they are embedded submanifolds – and it is possible to strengthen these necessary conditions for optimality in the sense of specifying the measures further. We formulate the conditions for a particular case of common interest.

We consider an optimal control problem in *Mayer form* (i.e., $L \equiv 0$) for a single-input control linear system with dynamics

$$\dot{x} = F(t, x, u) = f(t, x) + u g(t, x)$$

and the control set U a compact interval, $U = [a, b]$. Adjoining time as extra state variable, $\dot{t} \equiv 1$, and defining

$$F_0(t, x) = \begin{pmatrix} 1 \\ f(t, x) \end{pmatrix} \text{ and } G(t, x) = \begin{pmatrix} 0 \\ g(t, x) \end{pmatrix},$$

for a continuously differentiable function $k : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$, the expressions

$$\mathcal{L}_{F_0} k : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n,$$

$$(t, x) \mapsto (\mathcal{L}_{F_0} k)(t, x)$$

$$= \frac{\partial k}{\partial t}(t, x) + \frac{\partial k}{\partial x}(t, x) f(t, x)$$

and

$$\mathcal{L}_G k : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n,$$

$$(t, x) \mapsto (\mathcal{L}_G k)(t, x) = \frac{\partial k}{\partial x}(t, x) g(t, x)$$

represent the Lie (or directional) derivatives of the function k along the vector fields F_0 and G, respectively. In terms of this notation, the derivative of the function h_α (defining the manifold M_α) along trajectories of the system is given by

$$\dot{h}_\alpha(t, x(t)) = \frac{d}{dt} h_\alpha(t, x(t))$$

$$= \mathcal{L}_{F_0} h_\alpha(t, x(t)) + u(t) \mathcal{L}_G h_\alpha(t, x(t)).$$

If the function $\mathcal{L}_G h_\alpha$ does not vanish at a point $(\tilde{t}, \tilde{x}) \in M_\alpha$, then there exists a neighborhood V of $(\tilde{t}, \tilde{x})$ such that there exists a unique

control $u_\alpha = u_\alpha(t, x)$ which solves the equation $\dot{h}_\alpha(t, x) = 0$ on V and u_α is given in feedback form as

$$u_\alpha(t, x) = -\frac{\mathcal{L}_{F_0} h_\alpha(t, x)}{\mathcal{L}_G h_\alpha(t, x)}.$$

The manifold M_α is said to be *control invariant* of *relative degree* 1 if the Lie derivative of h_α with respect to G, $\mathcal{L}_G h_\alpha$, does not vanish anywhere on M_α and if the function $u_\alpha(t, x)$ is admissible, i.e., takes values in the control set $[a, b]$.

Thus, for a control-invariant submanifold of relative degree 1, the control that keeps the manifold invariant is unique, and the corresponding dynamics induce a unique flow on the constraint. This assumption corresponds to the least degenerate, i.e., in some sense most generic or common, scenario and is satisfied for many practical problems.

Suppose the reference extremal is normal and let Γ_α be an M_α-boundary arc defined over an open interval I with corresponding boundary control u_α that takes values in the interior of the control set along Γ_α. Then the Radon measure μ_α is absolutely continuous with respect to Lebesgue measure on I with continuous and nonnegative Radon-Nikodym derivative $\nu_\alpha(t)$ given by

$$\nu_\alpha(t) = \frac{\lambda(t)\left(\frac{\partial g}{\partial t}(t, x_*(t)) + [f, g](t, x_*(t))\right)}{\mathcal{L}_G h_\alpha(t, x_*(t))}$$

where $[f, g]$ denotes the Lie bracket of the time-varying vector fields f and g in the variable x,

$$[f, g](t, x) = \frac{\partial g}{\partial x}(t, x) f(t, x) - \frac{\partial f}{\partial x}(t, x) g(t, x).$$

In particular, in this case, the adjoint equation can be expressed in the more common form

$$\dot{\lambda}(t) = -\lambda(t)\frac{\partial F}{\partial x}(t, x_*, u_*) - \nu_\alpha(t)\frac{\partial h_\alpha}{\partial x}(t, x_*),$$

with all partial derivatives evaluated along the reference trajectory. Furthermore, the multiplier λ remains continuous at entry or exit if the controlled trajectory (x_*, u_*) meets the constraint

M_α transversally (e.g., see Schättler 2006). This follows from the following characterization of transversal connections between interior and boundary arcs due to Maurer (1977): if τ is an entry or exit junction time between an interior arc and an M_α-boundary arc for which the reference control u_* has a limit at τ along the interior arc, then the interior arc is transversal to M_α at entry or exit if and only if the control u_* is discontinuous at τ.

Informal Formulation of Necessary Conditions

In order to ensure the practicality of necessary conditions for optimality, it is essential that besides atomistic structures at junctions that lead to computable jumps in the multipliers, the Radon measures μ_α have no singular parts with respect to Lebesgue measure. If it is *assumed* a priori that optimal controlled trajectories are finite concatenations of interior and boundary arcs, and if the constraint sets have a reasonably regular structure (embedded submanifolds and transversal intersections thereof) and satisfy a rather technical *constraint qualification* (see Hartl et al. 1995) that guarantees that the restrictions of the system to active constraints have solutions, then it is possible to specify the above necessary conditions further and formulate more user friendly versions for the determination of the multipliers. Such formulations have become the standard for numerical computations, but they still have not always been established rigorously and somewhat carry the stigma of a heuristic nature. Nevertheless, it is often this more concrete set of conditions that allow to solve problems numerically and analytically. If then, in conjunction with sufficient conditions for optimality, it is possible to verify the optimality of the computed extremal solutions, this generates a satisfactory theoretical procedure. Such conditions, following Hartl et al. (1995), generally are referred to as the "informal theorem".

Suppose (x_*, u_*) is a normal extremal controlled trajectory defined over the interval $[t_0, T]$ with the property that the graph of x_* is a finite concatenation of interior and boundary arcs with junction times τ_i, $i = 1, \ldots, k$, $t_0 = \tau_0 < \tau_1 <$

$\ldots < \tau_k < \tau_{k+1} = T$. Under an appropriate constraint qualification, there exist a multiplier λ, $\lambda : [t_0, T] \to (\mathbb{R}^n)^*$, which is absolutely continuous on each subinterval $[\tau_i, \tau_{i+1}]$; multipliers $\nu_\alpha, \nu_\alpha : [t_0, T] \to (\mathbb{R}^n)^*$, which are continuous on each interval $[\tau_i, \tau_{i+1}]$; a vector $\mu \in (\mathbb{R}^k)^*$; and vectors $\eta(\tau_i) \in (\mathbb{R}^r)^*$, $i = 1, \ldots, k$, with nonnegative entries such that:

(a) (adjoint equation) On each interval (τ_i, τ_{i+1}), $i = 0, \ldots, r$, λ satisfies the adjoint equation in the form

$$\dot{\lambda}(t) = -\frac{\partial L}{\partial x}(t, x_*(t), u_*(t))$$
$$-\lambda(t)\frac{\partial F}{\partial x}(t, x_*(t), u_*(t))$$
$$-\sum_{\alpha=1}^{r} \nu_\alpha(t)\frac{\partial h_\alpha}{\partial x}(t, x_*),$$

with $\nu_\alpha(t) = 0$ if the constraint M_α is not active at time t. Assuming that no state space constraint is active at the terminal time, the value of the multiplier λ at the terminal time is given by the transversality condition

$$\lambda(T) = \frac{\partial \Phi}{\partial x}(T, x_*(T)) + \mu\frac{\partial \Psi}{\partial x}(T, x_*(T)).$$

At any junction time τ_i between an interior arc and a boundary arc, the multiplier λ may be discontinuous satisfying a jump condition of the form

$$\lambda(\tau_i-) = \lambda(\tau_i+) + \eta(\tau_i)\frac{\partial h}{\partial x}(\tau_i, x_*(\tau_i))$$

and the complementary slackness condition

$$\eta(\tau_i)\frac{\partial h}{\partial x}(\tau_i, x_*(\tau_i)) = 0$$

holds.

(b) The optimal control minimizes the Hamiltonian over the control set U along $(\lambda(t), x_*(t))$:

$$H(t, \lambda(t), x_*(t), u_*(t))$$
$$= \min_{v \in U} H(t, \lambda(t), x_*(t), v)$$

and at the junction times τ_i we have that

$$H(\tau_i, \lambda(\tau_i-), x_*(\tau_i), u_*(\tau_i-))$$
$$= H(\tau_i, \lambda(\tau_i+), x_*(\tau_i), u_*(\tau_i+))$$
$$-\eta(\tau_i)\frac{\partial h}{\partial t}(\tau_i, x_*(\tau_i)).$$

Sufficient Conditions for Optimality

The literature on sufficient conditions for optimality for optimal control problems with state space constraints is limited. The value function for an optimal control problem at a point (t, x) in the extended state space, $V = V(t, x)$, is defined as the infimum over all admissible controls u for which the corresponding trajectory starts at the point x at time t and satisfies all the constraints of the problem,

$$V(t, x) = \inf_{u \in \mathcal{U}} J(u).$$

Any sufficiency theory for optimal control problems, one way or another, deals with the solution of the corresponding Hamilton-Jacobi-Bellman (HJB) equation:

$$\frac{\partial V}{\partial t}(t, x) + \min_{u \in U}\left\{\frac{\partial V}{\partial x}(t, x)F(t, x, u)\right.$$
$$\left. + L(t, x, u)\right\} \equiv 0,$$
$$V(T, x) = \Phi(T, x) \text{ whenever } \Psi(T, x) = 0.$$

Value functions for optimal control problems rarely are differentiable everywhere, but generally have singularities along lower-dimensional submanifolds. Nevertheless, under some technical assumptions and with proper interpretations of the derivatives, this equation describes the evolution of the value function of an optimal control problem and, if an appropriate solution can be constructed, indeed solves the optimal control problem.

There exists a broad theory of viscosity solutions to the HJB equation (e.g., Fleming and

Soner 2005; Bardi and Capuzzo-Dolcetta 2008) that is also applicable to problems with state space constraints (Soner 1986) and, under varying technical assumptions, characterizes the value function V as the unique viscosity solution to the HJB equation. This has led to the development of algorithms that can be used to compute numerical solutions.

A more classical and more geometric approach to solving the HJB equation is based on the method of characteristics and goes back to the work of Boltyansky on a regular synthesis for optimal control problems without state space constraints (Boltyanskii 1966). This work follows classical ideas of fields of extremals from the calculus of variations and imposes technical conditions that allow to handle the singularities that arise in the value functions (e.g., see, Schättler and Ledzewicz 2012). Stalford's results in Stalford (1971) follow this approach for problems with state space constraints, but a broadly applicable theory of

regular synthesis, as it was developed by Piccoli and Sussmann in (2000) for problems without state space constraints, does not yet exist for problems with state space constraints. Results that embed a controlled reference extremal into a local field of extremals have been given by Bonnard et al. (2003) or Schättler (2006), and these constructions show the applicability of the concepts of a regular synthesis to problems with state space constraints as well.

Examples of Local Embeddings of Boundary Arcs

We illustrate the typical, i.e., in some sense most common, generic structures of local embeddings of boundary arcs in Figs. 1 and 2. The state constraint M_α is a control-invariant submanifold of relative degree 1 and represented by a horizontal line as it arises when limits on the size of a particular state are imposed. Figure 1 shows the typical entry-boundary-exit concatenations of an interior arc followed by a boundary arc and

Optimal Control with State Space Constraints, Fig. 1 A typical local synthesis around a boundary arc when no terminal constraints are present
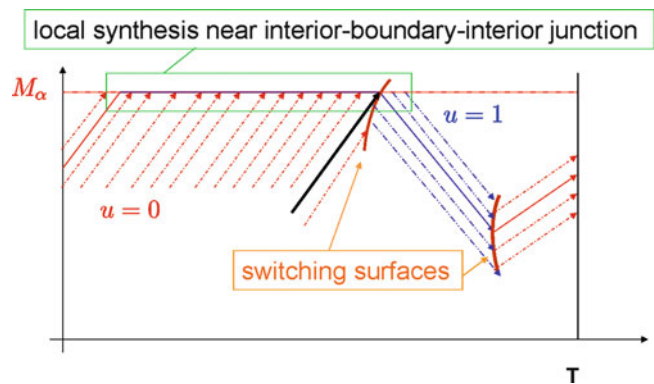
Optimal Control with State Space Constraints, Fig. 2 A typical local synthesis around a boundary arc when terminal constraints are present
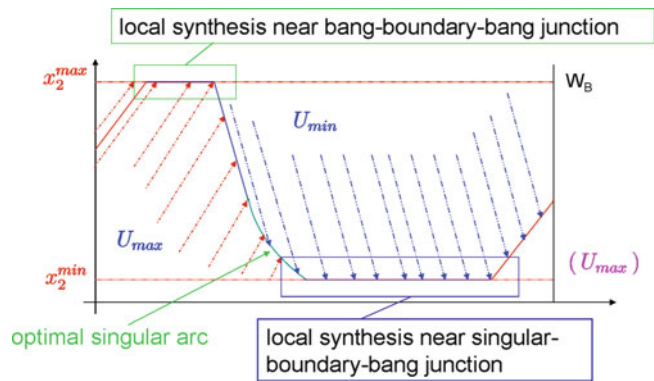
another interior arc. The local embedding of the boundary arc differs substantially from classical local imbeddings for unconstrained problems in the sense that this field necessarily contains small pieces of trajectories which, when propagated backward, are not close to the reference trajectory. This, however, does not affect the memoryless properties required for a synthesis forward in time, and strong local optimality of the reference trajectory can be proven combining synthesis type arguments with homotopy type approximations of the synthesis (Schättler 2006). The one trajectory marked as black line in Fig. 1 corresponds to an optimal trajectory that meets the constraint only at the junction point and immediately bounces back into the interior. Such a trajectory arises as the limit when the concatenation structure of optimal controlled trajectories changes from interior-boundary-interior arcs to trajectories that do not meet the constraint. These structures are one of the extra sources for singularities in the value function that come up in optimal control problems with state space constraints. Switching surfaces for the interior arcs, as one is also shown in this figure, do not cause such a loss of differentiability if they are crossed transversally be the extremal trajectories of the field.

Figure 2 depicts the structure of an optimal synthesis for a problem from electronics, the problem of minimizing the base transit time of bipolar homogeneous transistors. The electrical field that determines the transit time is controlled by tailoring a distribution of dopants in the base region, and this dopant profile becomes an important design parameter determining the speed of the device. But due to physical and engineering limitations, the variables describing the dopants need to be limited, and thus this becomes an optimal control problem with state space constraints represented by hard limits on the variables. The constraints here are control invariant of relative degree 1. Optimal solutions, in the presence of initial and terminal constraints, have both portions along the upper and lower control limits of the constrained variable and typically proceed from the upper to the lower values along an optimal singular control (which takes values in

the interior of the control set) in the interior of the admissible domain, possibly with saturation if the control limits are reached.

Cross-References

▶ Numerical Methods for Nonlinear Optimal Control Problems
▶ Optimal Control and Pontryagin's Maximum Principle

Bibliography

Bardi M, Capuzzo-Dolcetta I (2008) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Springer, New York

Boltyanskii VG (1966) Sufficient conditions for optimality and the justification of the dynamic programming principle. SIAM J Control Optim 4:326–361

Bonnard B, Faubourg L, Launay G, Trélat E (2003) Optimal control with state space constraints and the space shuttle re-entry problem. J Dyn Control Syst 9:155–199

Cesari L (1983) Optimization – theory and applications. Springer, New York

Fleming WH, Soner HM (2005) Controlled Markov processes and viscosity solutions. Springer, New York

Frankowska H (2006) Regularity of minimizers and of adjoint states in optimal control under state constraints. Convex Anal 13:299–328

Hartl RF, Sethi SP, Vickson RG (1995) A survey of the maximum principles for optimal control problems with state constraints. SIAM Rev 37:181–218

Ioffe AD, Tikhomirov VM (1979) Theory of extremal problems. North-Holland, Amsterdam/New York

Maurer H (1977) On optimal control problems with bounded state variables and control appearing linearly. SIAM J Control Optim 15:345–362

Piccoli B, Sussmann H (2000) Regular synthesis and sufficient conditions for optimality. SIAM J Control Optim 39:359–410

Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1962) Mathematical theory of optimal processes. Wiley-Interscience, New York

Rinaldi P, Schättler H (2003) Minimization of the base transit time in semiconductor devices using optimal control. In: Feng W, Hu S, Lu X (eds) Dynamical systems and differential equations. Proceedings of the 4th international conference on dynamical systems and differential equations. Wilmington, May 2002, pp 742–751

Schättler H (2006) A local field of extremals for optimal control problems with state space constraints of relative degree 1. J Dyn Control Syst 12:563–599

Schättler H, Ledzewicz U (2012) Geometric optimal control. Springer, New York

Soner HM (1986) Optimal control with state-space constraints I. SIAM J Control Optim 24:552–561

Stalford H (1971) Sufficient conditions for optimal control with state and control constraints. J Optim Theory Appl 7:118–135

Vinter RB (2000) Optimal control. Birkhäuser, Boston

Optimal Deployment and Spatial Coverage

Sonia Martínez

Department of Mechanical and Aerospace Engineering, University of California, La Jolla, San Diego, CA, USA

Abstract

Optimal deployment refers to the problem of how to allocate a finite number of resources over a spatial domain to maximize a performance metric that encodes certain quality of service. Depending on the deployment environment, the type of resource, and the metric used, the solutions to this problem can greatly vary.

Keywords

Coverage control algorithms; Facility location problems

Introduction

The problem of deciding what are optimal geographic locations to place a set of facilities has a long history and is the main subject in operations research and management science; see Drezner (1995). A facility can be broadly understood as a service such as a school; a hospital; an airport; an emergency service, such as a fire station; or, more generally, routes of a vehicle, from buses to aircraft, an autonomous vehicle, or a mobile sensor.

The specific formulation of facility location problems depends very much on the particular underlying application. A distinguishing feature is that all involve strategic planning, accounting for the long-term impact on the facility operating cost and their fast response to the demand. Thus, these problems lead to constrained optimization formulations which are typically very hard to solve optimally. The computational complexity of such problems, which, even in their most basic formulations, typically lead to NP-hard problems, has made their solution largely intractable until the advent of high-speed computing.

Locational optimization techniques have also been employed to solve optimal estimation problems by static sensor networks, mesh and grid optimization design, clustering analysis, data compression, and statistical pattern recognition; see Du et al. (1999). However, these solutions typically require centralized computations and availability of information at all times.

When the facilities are multiple vehicles or mobile sensors, the underlying dynamics may require additional changes and further analysis that guarantee the overall system stability. In what follows, we review a particular coverage control problem formulation in terms of the so-called expected-value multicenter functions that makes the analysis tractable leading to robust, distributed algorithm implementations employing computational geometric objects such as Voronoi partitions.

Basic Ingredients from Computational Geometry

In order to formulate a basic optimal deployment problem and algorithm, we require of several notions from computational geometry; see Bullo et al. (2009) for more information.

Let S be a measurable set of $\mathbb{R}^m$, for $m \in \mathbb{N}$, consider a distance function d on $\mathbb{R}^m$, and let $P = \{p_1, \ldots, p_n\}$ be n distinct points of S, corresponding to *locations* of certain facilities. The *Voronoi partition* of S generated by P and associated with d is given by $\mathcal{V}(P) = \{V_1(P), \ldots, V_n(P)\}$, where

$$V_i(P) = \{q \in S \mid d(p_i, q) \leq d(p_j, q),$$
$$j \in P \setminus \{i\}\}, \quad i \in \{1, \ldots, n\}.$$

Given $r \in \mathbb{R}_{>0}$, denote by $\overline{B}(p_i, r)$ the closed ball of center p_i and radius r. The *r-limited Voronoi partition* of S generated by P and associated with d is the Voronoi partition of the set $S \cap \cup_{i=1}^{n} \overline{B}(p_i, r)$, denoted as $\mathcal{V}_r(P) = \{V_{1,r}(P), \ldots V_{n,r}(P)\}$.

Let $\phi : S \rightarrow \mathbb{R}_{\geq 0}$ be a measurable *density* function on S. The *area* and the *centroid* (or *center of mass*) of $W \subseteq S$ with respect to ϕ are the values

$$A_\phi(W) = \int_W \phi(q) dq,$$

$$\mathrm{CM}_\phi(W) = \frac{1}{A_\phi(W)} \int_W q \phi(q) dq.$$

We say that the set of distinct points P in S is a *centroidal Voronoi configuration* (resp., a *r-limited centroidal Voronoi configuration*) if each p_i is at the centroid of its own Voronoi cell. That is, $p_i = \mathrm{CM}_\phi(V_i(P))$, $i \in \{1, \ldots, n\}$ (resp., $p_i = \mathrm{CM}_\phi(V_{i,r}(P))$, and $i \in \{1, \ldots, n\}$). Voronoi partitions and centroidal Voronoi configurations help assess the distribution of locations in a spatial domain as we establish below.

A Voronoi partition induces a natural *proximity graph*, called the *Delaunay graph*, over the set of points P. We recall that a *graph G* is a pair $G = (V, E)$ where V is a set of n vertices and E is a set of ordered pair of *vertices*, $E \subset V \times V$, called *edge set*. A *proximity graph* is a graph function defined on the set S, which assigns a set of distinct points $P \subset S$ to a graph $G(P) = (P, E(P))$, where $E(P)$ is a function of the relative locations of the point set. Example graphs include the following:

1. The *r-disk graph*, $\mathcal{G}_{\mathrm{disk},r}$, for $r \in \mathbb{R}_{>0}$. Here, $(p_i, p_j) \in E_{\mathrm{disk},r}(P)$ if $d(p_i, p_j) \leq r$.
2. The *Delaunay graph*, $\mathcal{G}_{\mathrm{D}}$. We have $(p_i, p_j) \in E_{\mathrm{D}}(P)$ if $V_i(P) \cap V_j(P) \neq \emptyset$.
3. The *r-limited Delaunay graph*, $\mathcal{G}_{\mathrm{LD},r}$, for $r \in \mathbb{R}_{>0}$. Here, $(p_i, p_j) \in E_{\mathrm{LD},r}(P)$ if $V_{i,r}(P) \cap V_{i,r}(P) \neq \emptyset$.

Expected-Value Multicenter Functions

Facility location problems consist of spatially allocating a number of sites to provide certain quality of service. Problems of this class are formulated in terms of multicenter functions and, in particular, expected-value multicenter functions.

To define these, consider $\phi : S \rightarrow \mathbb{R}_{\geq 0}$ a density function over a bounded measurable set $S \subset \mathbb{R}^m$. One can regard ϕ as a function measuring the probability that some event takes place over the environment. The larger the value of $\phi(q)$, the more important the location q will have. We refer to a nonincreasing and piecewise continuously differentiable function $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, possibly with finite jump discontinuities, as a *performance function*.

Performance functions describe the utility of placing a node at a certain distance from a location in the environment. The smaller the distance, the larger the value of f, that is, the better the performance. For instance, in sensing problems, performance functions can encode the signal-to-noise ratio between a source with an unknown location and a sensor attempting to locate it. Without loss of generality, it can be assumed that $f(0) = 0$.

An *expected-value multicenter function* models the expected value of the coverage over any point in S provided by a set of points $p_1, \ldots, p_n$. Formally,

$$\mathcal{H}(p_1, \ldots, p_n) = \int_S \max_{i \in \{1, \ldots, n\}} f(\|q - p_i\|_2) \phi(q) dq, \tag{1}$$

where $\| \cdot \|_2$ denotes the 2-norm of $\mathbb{R}^m$. This definition can be understood as follows: consider the best coverage of $q \in S$ among those provided by each of the nodes $p_1, \ldots, p_n$, which corresponds to the value $\max_{i \in \{1, \ldots, n\}} f(\|q - p_i\|_2)$. Then, modulate the performance by the importance $\phi(q)$ of the location q. Finally, the infinitesimal sum of this quantity over the environment S gives rise to $\mathcal{H}(p_1, \ldots, p_n)$ as a measure of the overall coverage provided by $p_1, \ldots, p_n$.

From here, we can formulate the following geometric optimization problem, known

as the *continuous p-median problem*, see Drezner (1995):

$$\max_{\{p_1,\dots,p_n\}\subset S} \mathcal{H}(p_1,\dots,p_n). \qquad (2)$$

The expected-value multicenter function can be alternatively described in terms of the Voronoi partition of S generated by $P = \{p_1,\dots,p_n\}$. Let us define the set

$$\mathcal{C} = \big\{(p_1,\dots,p_n) \in (\mathbb{R}^m)^n \mid p_i = p_j$$
$$\text{for some } i \neq j\big\},$$

consisting of tuples of n points, where some of them are repeated. Then, for $(p_1,\dots,p_n) \in S^n \setminus \mathcal{C}$, one has

$$\mathcal{H}(p_1,\dots,p_n) = \sum_{i=1}^{n} \int_{V_i(P)} f(\|q - p_i\|_2)\phi(q)dq. \qquad (3)$$

This expression of $\mathcal{H}$ is appealing because it clearly shows the result of the overall coverage of the environment as the aggregate contribution of all individual nodes. If $(p_1,\dots,p_n) \in \mathcal{C}$, then a similar decomposition of $\mathcal{H}$ can be written in terms of the distinct points $P = \{p_1,\dots,p_n\}$.

Inspired by (3), a more general version of the expected-value multicenter function is given next. Given $(p_1,\dots,p_n) \in S^n$ and a partition $\{W_1,\dots,W_n\}$ of S, let

$$\mathcal{H}(p_1,\dots,p_n, W_1,\dots,W_n)$$
$$= \sum_{i=1}^{n} \int_{W_i} f(\|q - p_i\|_2)\phi(q)dq. \qquad (4)$$

For all $(p_1,\dots,p_n) \in S^n \setminus \mathcal{C}$, we have that $\mathcal{H}(p_1,\dots,p_n) = \mathcal{H}(p_1,\dots,p_n, V_1(P),\dots,V_n(P))$. With respect to, e.g., sensor networks, this function evaluates the performance associated with an assignment of the sensors' locations at $(p_1,\dots,p_n)$ and a region assignment $(W_1,\dots,W_n)$.

Moreover, one can establish that the Voronoi partition (Du et al. 1999) $\mathcal{V}(P)$ is optimal for $\mathcal{H}$ among all partitions of S. That is, let $P =$

$\{p_1,\dots,p_n\} \in S$. For any performance function f and for any partition $\{W_1,\dots,W_n\}$ of S,

$$\mathcal{H}(p_1,\dots,p_n, V_1(P),\dots,V_n(P)) \geq$$
$$\mathcal{H}(p_1,\dots,p_n, W_1,\dots,W_n),$$

with a strict inequality if any set in $\{W_1,\dots,W_n\}$ differs from the corresponding set in $\{V_1(P),\dots,V_n(P)\}$ by a set of positive measure.

Next, we characterize the smoothness of the expected-value multicenter function (Cortés et al. 2005). Before stating the precise properties, let us introduce some useful notation. For a performance function f, let discont(f) denote the (finite) set of points where f is discontinuous. For each $a \in \text{discont}(f)$, define the limiting values from the left and from the right, respectively, as

$$f_-(a) = \lim_{x\to a^-} f(x), \qquad f_+(a) = \lim_{x\to a^+} f(x).$$

Recall that the line integral of a function $g : \mathbb{R}^2 \to \mathbb{R}$ over a curve C parameterized by a continuous and piecewise continuously differentiable map $\gamma : [0,1] \to \mathbb{R}^2$ is defined as follows:

$$\int_C g = \int_C g(\gamma)d\gamma := \int_0^1 g(\gamma(t))\,\|\dot{\gamma}(t)\|_2\,dt,$$

and is independent of the selected parameterization.

Now, given a set $S \subset \mathbb{R}^m$ that is bounded and measurable, a density $\phi : S \to \mathbb{R}_{\geq 0}$, and a performance function $f : \mathbb{R} \to_{\geq 0} \mathbb{R}$, the expected-value multicenter function $\mathcal{H} : S^n \to \mathbb{R}$ is globally Lipschitz (Given $S \subset \mathbb{R}^h$, a function $f : S \to \mathbb{R}^k$ is globally Lipschitz if there exists $K \in \mathbb{R}_{>0}$ such that $\|f(x) - f(y)\|_2 \leq K\|x - y\|_2$ for all $x, y \in S$.) on S^n; and continuously differentiable on $S^n \setminus \mathcal{C}$, where for $i \in \{1,\dots,n\}$

$$\frac{\partial\mathcal{H}}{\partial p_i}(P) = \int_{V_i(P)} \frac{\partial}{\partial p_i} f(\|q - p_i\|_2)\phi(q)dq$$
$$+ \sum_{a\in\text{discont}(f)} (f_-(a) - f_+(a))$$
$$\int_{V_i(P)\cap\partial\overline{B}(p_i,a)} \mathrm{n}_{\text{out}}(q)\phi(q)dq, \qquad (5)$$

where n_{out} is the outward normal vector to $\overline{B}(p_i, a)$.

Different performance functions lead to different expected-value multicenter functions. Let us examine some important cases.

Distortion Problem

Consider the performance function $f(x) = -x^2$. Then, on $S^n \setminus \mathcal{C}$, the expected-value multicenter function takes the form

$$\mathcal{H}_{\text{distor}}(p_1, \ldots, p_n) = -\sum_{i=1}^{n} \int_{V_i(P)} \|q - p_i\|_2^2 \phi(q) dq.$$

In signal compression $-\mathcal{H}_{\text{distor}}$ is referred to as the *distortion function* and is relevant in many disciplines where including vector quantization, signal compression, and numerical integration; see Gray and Neuhoff (1998) and Du et al. (1999). Here, distortion refers to the average deformation (weighted by the density ϕ) caused by reproducing $q \in S$ with the location p_i in $P = \{p_1, \ldots, p_n\}$ such that $q \in V_i(P)$. By means of the Parallel Axis Theorem (see Hibbeler 2006), it is possible to express $\mathcal{H}_{\text{distor}}$ as a sum

$$\mathcal{H}_{\text{distor}}(p_1, \ldots, p_n, W_1, \ldots, W_n)$$
$$= \sum_{i=1}^{n} -J_\phi(W_i, \text{CM}_\phi(W_i))$$
$$- A_\phi(W_i) \|p_i - \text{CM}_\phi(W_i)\|_2^2, \quad (6)$$

where $J_\phi(W, p) = \int_W \|q - p\|^2 \phi(q) dq$ is the so-called moment of inertia of the region W about p with respect to ϕ. In this way, the terms $J_\phi(W_i, \text{CM}_\phi(W_i))$ only depend on the partition of S, whereas the second terms multiplied by $A_\phi(W_i)$ include the particular location of the points. As a consequence of this observation, the optimality of the centroid locations for $\mathcal{H}_{\text{distor}}$ follows Bullo et al. (2009). More precisely, let $\{W_1, \ldots, W_n\}$ be a partition of S. Then, for any set points $P = \{p_1, \ldots, p_n\}$ in S,

$$\mathcal{H}_{\text{distor}}(\text{CM}_\phi(W_1), \ldots, \text{CM}_\phi(W_n), W_1, \ldots, W_n)$$
$$\geq \mathcal{H}_{\text{distor}}(p_1, \ldots, p_n, W_1, \ldots, W_n),$$

and the inequality is strict if there exists $i \in \{1, \ldots, n\}$ for which W_i has nonvanishing area and $p_i \neq \text{CM}_\phi(W_i)$. In other words, the centroid locations $\text{CM}_\phi(W_1), \ldots, \text{CM}_\phi(W_n)$ are optimal for $\mathcal{H}_{\text{distor}}$ among all configurations in S.

Note that when $n = 1$, the node location that optimizes $p \mapsto \mathcal{H}_{\text{distor}}(p)$ is the centroid of the set S, denoted by $\text{CM}_\phi(S)$.

Recall that the gradient of $\mathcal{H}_{\text{distor}}$ on $S^n \setminus \mathcal{C}$ takes the form,

$$\frac{\partial \mathcal{H}_{\text{distor}}}{\partial p_i}(P) = 2A_\phi(V_i(P))(\text{CM}_\phi(V_i(P)) - p_i),$$
$$i \in \{1, \ldots, n\},$$

that is, the ith component of the gradient points in the direction of the vector going from p_i to the centroid of its Voronoi cell. The critical points of $\mathcal{H}_{\text{distor}}$ are therefore the set of centroidal Voronoi configurations in S. This is a natural generalization of the result for the case $n = 1$, where the optimal node location is the centroid $\text{CM}_\phi(S)$.

Area Problem

For $r \in \mathbb{R}_{>0}$, consider the performance function $f(x) = 1_{[0,r]}(x)$, that is, the indicator function of the closed interval $[0, r]$. Then, the expected-value multicenter function becomes

$$\mathcal{H}_{\text{area},r}(p_1, \ldots, p_n) = \sum_{i=1}^{n} A_\phi(V_i(P) \cap \overline{B}(p_i, r))$$
$$= A_\phi(\cup_{i=1}^{n} \overline{B}(p_i, r)),$$

which corresponds to the area, measured according to ϕ, covered by the union of the n balls $\overline{B}(p_1, r), \ldots, \overline{B}(p_n, r)$.

Let us see how the computation of the partial derivatives of $\mathcal{H}_{\text{area},r}$ specializes in this case. Here, the performance function is differentiable everywhere except at a single discontinuity, and its derivative is identically zero. Therefore, the first term in (5) vanishes. The gradient of $\mathcal{H}_{\text{area},r}$

on $S^n \setminus \mathcal{C}$ then takes the form, for each $i \in \{1, \ldots, n\}$,

$$\frac{\partial \mathcal{H}_{\text{area},r}}{\partial p_i}(P) = \int_{V_i(P) \cap \partial \overline{B}(p_i, r)} \mathrm{n}_{\text{out}}(q)\phi(q)dq,$$

where n_{out} is the outward normal vector to $\overline{B}(p_i, r)$. The critical points of $\mathcal{H}_{\text{area},r}$ correspond to configurations with the property that each p_i is a local maximum for the area of $V_{i,r}(P) = V_i(P) \cap \overline{B}(p_i, r)$ at fixed $V_i(P)$. We refer to these configurations as *r-limited area-centered* Voronoi configurations.

Optimal Deployment Algorithms

Once a set of optimal deployment configurations have been characterized, the next step is to devise a distributed algorithm that allows a group of mobile robots to converge to such configurations. Gradient algorithms are the first of the options that should be explored.

For the expected-value multicenter functions, robots whose dynamics can be described by first-order integrator dynamics and which can communicate at predetermined *communication rounds* of a fixed time schedule, these laws present a similar structure, loosely described as follows:

> [*Informal description*] In each communication round, each robot performs the following tasks: (i) it transmits its position and receives its neighbors' positions; (ii) it computes a notion of the geometric center of its own cell, determined according to some notion of partition of the environment. (iii) Between communication rounds, each robot moves toward this center.

The notions of geometric center and of partition of the environment differ depending on what is the type of expected-value multicenter function used. In the *Voronoi-center deployment algorithm*, the geometric center just reduces to $\mathrm{CM}_\phi(V_i)$. In the *limited-Voronoi-normal* deployment problem in (ii), each agent computes the direction of $v = \frac{\partial \mathcal{H}_{\text{area},r}}{\partial p_i}$ for some r and (iii) moves for a maximum step size in this direction to ensure the area function will be decreased.

The Voronoi-center deployment algorithm achieves convergence of a set of nodes to a centroidal Voronoi configuration, thus maximizing the expected-value multicenter function $\mathcal{H}_{\text{distor}}$. The algorithm is distributed over the proximity graph $\mathcal{G}_D$, as the computation of the centroids requires information in $\mathcal{N}_{\mathcal{G}_D}(p_i)$, for each $i \in \{1, \ldots, n\}$. Additional properties of this algorithm are that the algorithm is adaptive to agent departures or arrivals and amenable to asynchronous implementations.

On the other hand, the limited-Voronoi-normal deployment algorithm achieves convergence to a set that locally maximizes the area covered by the set of sensing balls. The algorithm is distributed in the sense that agents only need to know information from neighbors in the proximity graph $\mathcal{G}_{2r}$ or, more precisely, $\mathcal{G}_{\text{LD},r}$. Thus, it can be implemented by agents that employ range-limited interactions. It enjoys similar robustness properties as the Voronoi-center deployment algorithm.

Simulation Results

We show evolutions of the Voronoi-centroid deployment algorithm in Fig. 1. One can verify that the final network configuration is a centroidal Voronoi configuration. For each evolution we depict the initial positions, the trajectories, and the final positions of all robots.

Finally, we show an evolution of limited-Voronoi-normal deployment algorithm in Fig. 2. One can verify that the final network configuration is an $\frac{r}{2}$-limited area-centered Voronoi configuration. In other words, the deployment task is achieved.

Future Directions for Research

The algorithms described above achieve locally optimal deployment configurations with respect to expected-value multicenter functions. However, this simplified setting does not account for many important constraints, such as obstacles and deployment in non-convex environments (Pimenta et al. 2008; Caicedo-Nùñez and Žefran 2008), deployment with visibility sensors, range-limited and wedge-shaped
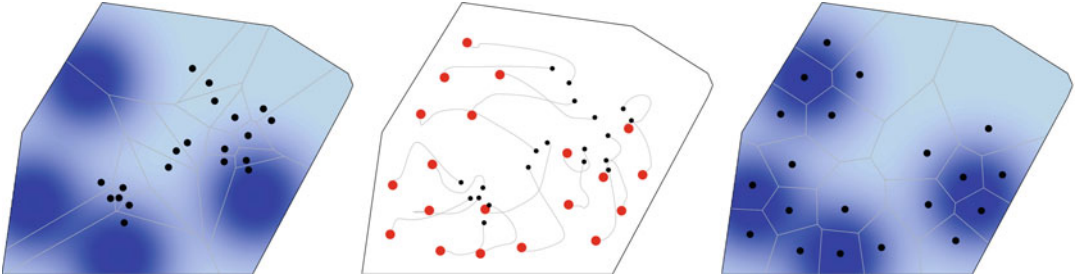
Optimal Deployment and Spatial Coverage, Fig. 1
The evolution of the Voronoi-centroid deployment algorithm with $n = 20$ robots. The *left-hand* (resp., *right-hand*) figure illustrates the initial (resp., final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. After 13 s, the value of $\mathcal{H}_{distor}$ has monotonically increased to approximately -0.515
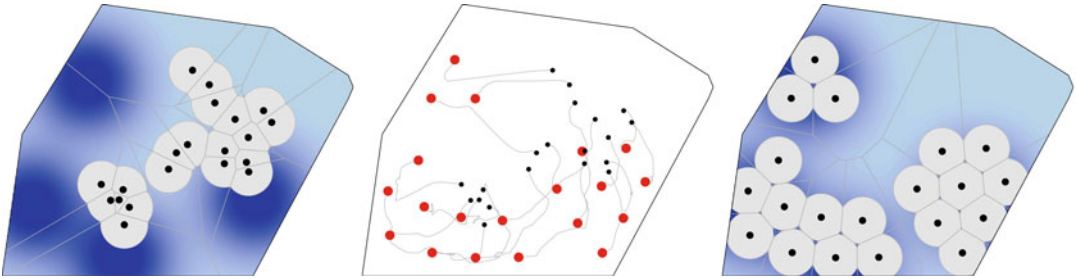
Optimal Deployment and Spatial Coverage, Fig. 2
The evolution of the limited-Voronoi-normal deployment algorithm with $n = 20$ robots and $r = 0.4$. The *left-hand* (resp., *right-hand*) figure illustrates the initial (respectively, final) locations and Voronoi partition. The central figure illustrates the evolution of the robots. The $\frac{r}{2}$-limited Voronoi cell of each robot is plotted in *light gray*. After 36 s, the value of $\mathcal{H}_{area}$, with $a = \frac{r}{2}$, has monotonically increased to approximately 14.141

footprints (Ganguli et al. 2006; Laventall and Cortés 2009), and energy and vehicle dynamical restrictions (Kwok and Martínez 2010a,b). Deployment strategies find application in exploration and data gathering tasks, and so these algorithms have been expanded to account for uncertainty and learning of unknown density functions (Schwager et al. 2009; Graham and Cortés 2012; Zhong and Cassandras 2011; Martínez 2010). Gossip and self-triggered communications (Bullo et al. 2012; Nowzari and Cortés 2012), self-triggered computations for region approximation (Ru and Martínez 2013), and area equitable partitions (Cortés 2010) have also been investigated. Much work is currently being devoted to solve on the current limitations of these nontrivial extensions, which make the problem settings significantly harder to solve.

Cross-References

▶ Graphs for Modeling Networked Interactions
▶ Multi-vehicle Routing
▶ Networked Systems

Bibliography

Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Applied mathematics series. Princeton University Press. Available at http://www.coordinationbook.info

Bullo F, Carli R, Frasca P (2012) Gossip coverage control for robotic networks: dynamical systems on the the space of partitions. SIAM J Control Optim 50(1): 419–447

Caicedo-Nùñez CH, Žefran M (2008) Performing coverage on nonconvex domains. In: IEEE conference on control applications, San Antonio, pp 1019–1024

Cortés J (2010) Coverage optimization and spatial load balancing by robotic sensor networks. IEEE Trans Autom Control 55(3):749–754

Cortés J, Martínez S, Bullo F (2005) Spatially-distributed coverage optimization and control with limited-range interactions. ESAIM Control Optim Calc Var 11:691–719

Drezner Z (ed) (1995) Facility location: a survey of applications and methods. Springer series in operations research. Springer, New York

Du Q, Faber V, Gunzburger M (1999) Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev 41(4):637–676

Ganguli A, Cortés J, Bullo F (2006) Distributed deployment of asynchronous guards in art galleries. In: American control conference, Minneapolis, pp 1416–1421

Graham R, Cortés J (2012) Cooperative adaptive sampling of random fields with partially known covariance. Int J Robust Nonlinear Control 22(5):504–534

Gray RM, Neuhoff DL (1998) Quantization. IEEE Trans Inf Theory 44(6):2325–2383. Commemorative Issue 1948–1998

Hibbeler R (2006) Engineering mechanics: statics & dynamics, 11th edn. Prentice Hall, Upper Saddle River

Kwok A, Martínez S (2010a) Deployment algorithms for a power-constrained mobile sensor network. Int J Robust Nonlinear Control 20(7):725–842

Kwok A, Martínez S (2010b) Unicycle coverage control via hybrid modeling. IEEE Trans Autom Control 55(2):528–532

Laventall K, Cortés J (2009) Coverage control by multi-robot networks with limited-range anisotropic sensory. Int J Control 82(6):1113–1121

Martínez S (2010) Distributed interpolation schemes for field estimation by mobile sensor networks. IEEE Trans Control Syst Technol 18(2):491–500

Nowzari C, Cortés J (2012) Self-triggered coordination of robotic networks for optimal deployment. Automatica 48(6):1077–1087

Pimenta L, Kumar V, Mesquita R, Pereira G (2008) Sensing and coverage for a network of heterogeneous robots. In: IEEE international conference on decision and control, Cancun, pp 3947–3952

Ru Y, Martínez S (2013) Coverage control in constant flow environments based on a mixed energy-time metric. Automatica 49:2632–2640

Schwager M, Rus D, Slotine J (2009) Decentralized, adaptive coverage control for networked robots. Int J Robot Res 28(3):357–375

Zhong M, Cassandras C (2011) Distributed coverage control and data collection with mobile sensor networks. IEEE Trans Autom Control 56(10):2445–2455

Optimal Sampled-Data Control

Yutaka Yamamoto
Department of Applied Analysis and Complex Dynamical Systems, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Abstract

This article gives a brief overview on the modern development of sampled-data control. Sampled-data systems intrinsically involve a mixture of two different time sets, one continuous and the other discrete. Due to this, sampled-data systems cannot be characterized in terms of the standard notions of transfer functions, steady-state response, or frequency response. The technique of lifting resolves this difficulty and enables the recovery of such concepts and simplified solutions to sampled-data H^∞ and H^2 optimization problems. We review the lifting point of view, its application to such optimization problems, and finally present an instructive numerical example.

Keywords

Computer control; Frequency response; H^∞ and H^2 optimization; Lifting; Transfer operator

Introduction

A sampled-data control system consists of a continuous-time plant and a discrete-time controller, with sample and hold devices that serve as an interface between these two components. As can be seen from this fact, sampled-data systems are *not* time invariant, and various problems arise from this property.

To be more specific, consider the unity-feedback control system shown in Fig. 1; r is the reference signal, y the system output, and e the error signal. These are continuous-time signals. The error $e(t)$ goes through the *sampler*
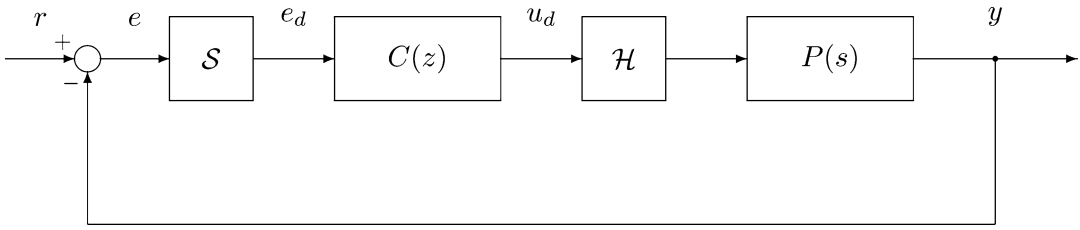
Optimal Sampled-Data Control, Fig. 1 A unity-feedback system
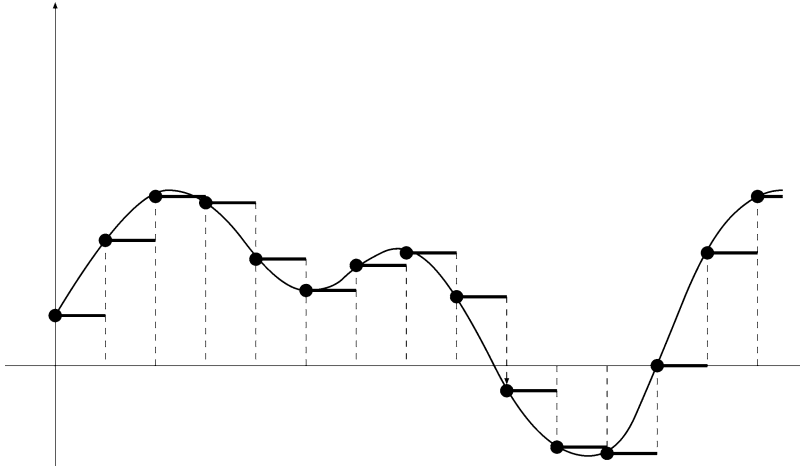
Optimal Sampled-Data Control, Fig. 2 Sampling with 0-order hold

(or an *A/D converter*) $\mathcal{S}$. This sampler reads out the values of $e(t)$ at every time step h called the *sampling period* and produces a discrete-time signal $e_d[k]$, $k = 0, 1, 2, \ldots$ (Fig. 2). In particular, the sampling operator $\mathcal{S}$ acts on a continuous-time signal $w(t)$, $t \geq 0$, as

$$\mathcal{S}(w)[k] := w(kh), \quad k = 0, 1, 2, \ldots$$

The discretized signal is then processed by the discrete-time controller $C(z)$ and becomes a control input u_d. There can also be a quantization effect, although for the sake of simplicity this is neglected here. The obtained signal u_d then goes through another interface $\mathcal{H}$ called a *hold device* or a *D/A converter* to become a continuous-time signal. A typical example is the 0-*order hold* where $\mathcal{H}$ simply maintains the value of a discrete-time signal $w[k]$ constant as its output until the next sampling time:

$$(\mathcal{H}(w[k]))(t) := w[k], \text{ for } kh \leq t < (k+1)h.$$

A typical sample-hold action is shown in Fig. 2.

While one can consider a nonlinear plant P or controller C, or infinite-dimensional P and C we confine ourselves to linear and finite-dimensional P and C, and also suppose that P and C are time invariant in continuous time and in discrete time, respectively.

The Main Difficulty

As stated above, the unity-feedback system Fig. 1 is not time invariant either in continuous time or in discrete time, even when the plant and controller are both time invariant in their respective domains of operators. The mixture of the two time sets prohibits the total closed-loop system from being time invariant.

The lack of time-invariance implies that we cannot naturally associate to sampled-data systems such classical concepts of transfer functions, steady-state response and frequency response.

One can regard Fig. 1 as a time-invariant discrete-time system by ignoring the intersample behavior and focusing attention on the sample-point behavior only. But the obtained model does not then reflect what happens between sampling times. This approach can lead to the neglect of undesirable inter-sample oscillations, called *ripples*. To monitor the intersample behavior, the notion of the *modified z-transform* was introduced, see, e.g., Jury (1958) and Ragazzini and Franklin (1958); however, this transform is usable only *after the controller has been designed* and hence not for the design problems considered in this article.

Lifting: A Modern Approach

A new approach was introduced around 1990–1991 (Bamieh et al. 1991; Tadmor 1991; Toivonen 1992; Yamamoto 1990, 1994). The new idea, now called *lifting*, makes it possible to describe sampled-data systems via a *time-invariant model while maintaining the intersample behavior*.

Let $f(t)$ be a continuous-time signal. Instead of sampling $f(t)$, we will represent it as a *sequence* of *functions*. Namely, we set up the correspondence:

$$\mathcal{L} : f \mapsto \{f[k](\theta)\}_{k=0}^{\infty},$$
$$f[k](\theta) = f(kh + \theta), \quad 0 \le \theta < h \quad (1)$$

See Fig. 3.

This idea makes it possible to view a (time-invariant or even periodically time-varying) *continuous-time* system as a linear, *time-invariant discrete-time* system.

Let

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t). \tag{2}$$

be a given continuous-time plant and lift the input $u(t)$ to obtain $u[k](\cdot)$. We apply this lifted input with the timing $t = kh$ (h is the prespecified sampling rate as above) and observe how it affects the system. Let $x[k]$ be the state at time $t = kh$. The state $x[k + 1]$ at time $(k + 1)h$ is given by

$$x[k + 1] = e^{Ah}x[k] + \int_0^h e^{A(h-\tau)}Bu[k](\tau)d\tau. \tag{3}$$

The right-hand side integral defines an operator

$$L^2[0, h] \to \mathbb{R}^n : u(\cdot) \mapsto \int_0^h e^{A(h-\tau)}Bu(\tau)d\tau.$$

While the state-transition (3) only described a discrete-time update, the system keeps producing an output during the intersample period. If we consider the lifting of $x(t)$, it is easily seen to be described by

$$x[k](\theta) = e^{A\theta}x[k] + \int_0^\theta e^{A(\theta-\tau)}Bu[k](\tau)d\tau.$$

As such, the lifted output $y[k](\cdot)$ is given by

$$y[k](\theta) = Ce^{A\theta}x[k] + \int_0^\theta Ce^{A(\theta-\tau)}Bu[k](\tau)d\tau. \tag{4}$$

Observe that formulas (3) and (4) take the form

Optimal Sampled-Data Control, Fig. 3 Lifting

$$x[k + 1] = \mathcal{A}x[k] + \mathcal{B}u[k]$$

$$y[k] = \mathcal{C}x[k] + \mathcal{D}u[k],$$

and the operators $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ do not depend on the time variable k. In other words, it is possible to describe this continuous-time system with discrete timing, once we adopt the lifting point of view. To be more precise, the operators $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are defined as follows:

$$\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^n : x \mapsto e^{Ah}x$$
$$\mathcal{B} : L^2[0, h] \to \mathbb{R}^n : u \mapsto \int_0^h e^{A(h-\tau)}Bu(\tau)d\tau$$
$$\mathcal{C} : \mathbb{R}^n \to L^2[0, h] : x \mapsto Ce^{A(\theta)}x$$
$$\mathcal{D} : L^2[0, h] \to L^2[0, h] : u \mapsto$$
$$\int_0^\theta Ce^{A(\theta-\tau)}Bu(\tau)d\tau$$

(5)

Thus the continuous-time plant (2) can be described by a *time-invariant* discrete-time model. Once this is done, it is straightforward to connect this expression with a discrete-time controller, and hence, sampled-data systems (for example, Fig. 1) can be fully described by time-invariant discrete-time equations, without discarding the intersampling information. We will also denote the overall equation (with discrete-time controller included) abstractly in the form

$$x[k + 1] = \mathcal{A}x[k] + \mathcal{B}u[k]$$
$$y[k] = \mathcal{C}x[k] + \mathcal{D}u[k].$$

(6)

While the obtained discrete-time model is a time invariant, the input and output spaces are now infinite dimensional. Its *transfer function (operator)* is defined as

$$G(z) := \mathcal{D} + \mathcal{C}(zI - \mathcal{A})^{-1}\mathcal{B}.$$

(7)

Note that $\mathcal{A}$ in (6) is a matrix because it is so for $\mathcal{A}$ in (5). Hence, (6) is stable if $G(z)$ is analytic for $\{z : |z| \geq 1\}$, provided that there is no unstable pole-zero cancellation.

Definition 1 *Let $G(z)$ be the transfer operator of the lifted system given by (7), which is stable in the sense above. The* frequency response operator *is the operator*

$$G(e^{j\omega h}) : L^2[0, h] \to L^2[0, h]$$

(8)

regarded as a function of $\omega \in [0, \omega_s)$ ($\omega_s := 2\pi/h$). Its gain *at ω is defined to be*

$$\|G(e^{j\omega h})\| = \sup_{v \in L^2[0,h]} \frac{\|G(e^{j\omega h})v\|}{\|v\|}.$$

(9)

The maximum $\|G(e^{j\omega h})\|$ over $[0, \omega_s)$ is the H^∞ norm of $G(z)$. The H^2-norm of G is defined by

$$\|G\|_2 := \left(\frac{h}{2\pi} \int_0^{2\pi/h} trace \{G^*(e^{j\omega h})G(e^{j\omega h})\}d\omega \right)^{1/2},$$

(10)

where the trace *here is taken in the sense of Hilbert-Schmidt norm; see Chen and Francis (1995) for details.*

H^∞ and H^2 Control Problems

A significant consequence of the lifting approach described above is that various robust control problems such as H^∞ and H^2 control problems for sampled-data control systems can be converted to corresponding discrete-time (finite-dimensional) problems. The approach was initiated by Chen and Francis (1990) and later solved by Bamieh and Pearson (1992), Kabamba and Hara (1993), Sivashankar and Khargonekar (1994), Tadmor (1991), and Toivonen (1992) in more complete forms; see Chen and Francis (1995) for the pertinent historical accounts.

Let us introduce the notion of generalized plants. Suppose that a continuous time plant is given in the following model:

$$\dot{x}_c(t) = Ax_c(t) + B_1w(t) + B_2u(t)$$
$$z(t) = C_1x_c(t) + D_{11}w(t) + D_{12}u(t) \quad (11)$$
$$y(t) = C_2x_c(t)$$

Here w is the exogenous input, $u(t)$ control input, $y(t)$ measured output, and $z(t)$ is the controlled output. The objective is to design a controller that takes the sampled measurements of y and returns a control variable u according to the following formula:

$$x_d[k + 1] = A_dx_d[k] + B_d\mathcal{S}y[k]$$
$$v[k] = C_dx_d[k] + D_d\mathcal{S}y[k] \quad (12)$$
$$u[k](\theta) = H(\theta)v[k]$$
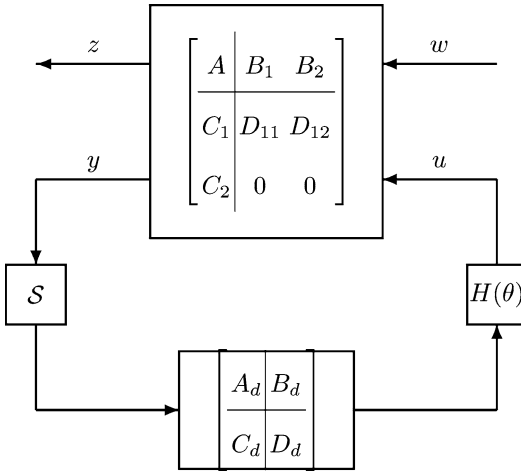
Optimal Sampled-Data Control, Fig. 4 Sampled feedback system

where $H(\theta)$ is a suitable hold function. This is depicted in Fig. 4. The objective here is to design or characterize a controller that achieves a prescribed performance level $\gamma > 0$ in such a way that

$$\|T_{zw}\|_\infty < \gamma \qquad (13)$$

where T_{zw} denotes the closed-loop transfer operator from w to z. This is the H^∞ control problem for sampled-data systems. If we take the H^2-norm (10) instead, then the problem becomes that of the H^2 (sub)optimal control problem.

The difficulty here is that both w and z are continuous-time variables, and hence their lifted variables are infinite dimensional. A remarkable fact here is that the H^∞ problem (and the H^2 problem as well) (13) can be equivalently transformed to an H^∞ problem for a *finite-dimensional* discrete-time system. We will indicate in the next section how this can be done.

H^∞ Norm Computation and Reduction to Finite Dimension

Let us write the system (11) and (12) in the form

$$\begin{aligned} x[k+1] &= \mathcal{A}x[k] + \mathcal{B}u[k] \\ y[k] &= \mathcal{C}x[k] + \mathcal{D}u[k]. \end{aligned} \qquad (14)$$

as in (6). For simplicity of treatments, assume D_{11} in (11) to be zero; for the general case, see Yamamoto and Khargonekar (1996).

Let $G(z)$ be the transfer operator $G(z) := \mathcal{D} + \mathcal{C}(zI - \mathcal{A})^{-1}\mathcal{B}$. The H^∞ norm of G is given as the maximum of the singular values of the gain $G(e^{j\omega h})$ for $\omega \in [0, 2\pi/h)$.

Now consider the singular value equation

$$(\gamma^2 I - G^*G(e^{j\omega h}))w = 0. \qquad (15)$$

and suppose that $\gamma > \|\mathcal{D}\|$. A crux here is that $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are finite-rank operators, and we can reduce this to a finite-dimensional rank condition. Taking the adjoint of (14), we obtain

$$\begin{aligned} p[k] &= \mathcal{A}^* p_{k+1} + \mathcal{C}^* v[k] \\ e[k] &= \mathcal{B}^* p_{k+1} + \mathcal{D}^* v[k]. \end{aligned}$$

Taking the z-transforms of both sides, setting $z = e^{j\omega h}$, and substituting $v = y$ and $e = \gamma^2 w$, we obtain

$$\begin{aligned} e^{j\omega h}x &= \mathcal{A}x + \mathcal{B}w \\ p &= e^{j\omega h}\mathcal{A}^* p + \mathcal{C}^*(\mathcal{C}x + \mathcal{D}w) \\ (\gamma^2 - \mathcal{D}^*\mathcal{D})w &= e^{j\omega h}\mathcal{B}^* p + \mathcal{D}^*\mathcal{C}x. \end{aligned}$$

Eliminating the variable w then yields

$$\left(e^{j\omega h} \begin{bmatrix} I & \mathcal{B}R_\gamma^{-1}\mathcal{B}^* \\ 0 & \mathcal{A}^* + \mathcal{C}^*\mathcal{D}R_\gamma^{-1}\mathcal{B}^* \end{bmatrix} - \begin{bmatrix} \mathcal{A} + \mathcal{B}R_\gamma^{-1}\mathcal{D}^*\mathcal{C} & 0 \\ \mathcal{C}^*(I + \mathcal{D}R_\gamma^{-1}\mathcal{D}^*)\mathcal{C} & I \end{bmatrix} \right) \begin{bmatrix} x \\ p \end{bmatrix} = 0 \qquad (16)$$

where $R_\gamma = (\gamma I - \mathcal{D}^*\mathcal{D})$. The important point to be noted here is that all the operators appearing here are actually matrices. For example, $\mathcal{B}$ is an

operator from $L^2[0, h]$ to $\mathbb{R}^n$, and its adjoint $\mathcal{B}^*$ is an operator from $\mathbb{R}^n$ to $L^2[0, h]$. Hence, the composition $\mathcal{B}R_\gamma^{-1}\mathcal{B}^*$ is a linear operator from

$\mathbb{R}^n$ into itself, i.e., a matrix. Thus, for a given γ the singular value equation admits a nontrivial solution w for (15) if and only if the *finite-dimensional equation* (16) admits a nontrivial solution $\begin{bmatrix} x & p \end{bmatrix}^T$ (Yamamoto 1993; Yamamoto and Khargonekar 1996). (Note that R_γ is invertible since $\gamma > \|\mathcal{D}\|$.)

It is possible to find matrices $\overline{A}, \overline{B}, \overline{C}$ such that $\overline{A} = \mathcal{A} + \mathcal{B}R_\gamma^{-1}\mathcal{D}^*\mathcal{C}$, $\overline{B}\,\overline{B}^*/\gamma^2 = \mathcal{B}R_\gamma^{-1}\mathcal{B}^*$, and $\overline{C}^*\overline{C} = \mathcal{C}^*(I + \mathcal{D}R_\gamma^{-1}\mathcal{D}^*)\mathcal{C}$, and hence (16) is equivalent to

$$\left(\lambda \begin{bmatrix} I & -\overline{B}\,\overline{B}^*/\gamma^2 \\ 0 & \overline{A}^* \end{bmatrix} - \begin{bmatrix} \overline{A} & 0 \\ -\overline{C}^*\overline{C} & I \end{bmatrix} \right) \begin{bmatrix} x \\ p \end{bmatrix} = 0 \tag{17}$$

for $\lambda = e^{j\omega h}$. In other words, we have that $\|G\|_\infty < \gamma$ if and only if there exists no λ of modulus 1 such that (17) holds.

It can be proven that by substituting the expressions of (11) and (12) for $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, one obtains a finite-dimensional discrete-time generalized plant G_d with digital controller (12) such that $\|G\|_\infty < \gamma$ if and only if $\|G_d\|_\infty < \gamma$. The precise formulas for the discrete-time plant can be found, e.g., Bamieh and Pearson (1992), Chen and Francis (1995), Kabamba and Hara (1993), Yamamoto and Khargonekar (1996), and Cantoni and Glover (1997).

An H^∞ Design Example

For sampled-data control systems, there used to be, and still is, a rather common myth that if one takes a sufficiently fast sampling rate, it will not cause a major problem. This can be true for continuous-time design, but we here show that if we employ a sample-point discretization without a performance consideration for intersampling behavior, fast sampling rates can cause a serious problem.

Take a simple second-order plant $P(s) = 1/(s^2 + 0.1s + 1)$, and consider the disturbance rejection problem minimizing the H^∞-norm from w to z as given in Fig. 5. Set the sampling time $h = 0.5$. We execute the following:

- Sampled-data H^∞ design with the generalized plant

$$G(s) = \begin{bmatrix} P(s) & P(s) \\ P(s) & P(s) \end{bmatrix},$$

- Discrete-time H^∞ design with the discrete-time generalized plant $G_d(z)$ given by the step-invariant transformation (see, e.g., Chen and Francis 1995) of $G(s)$.

Figures 6 and 7 show the frequency and time responses of the two resulting closed-loop systems, respectively. In Fig. 6, the solid curve shows the response of the sampled design, while the dash-dotted curve shows the discrete-time frequency response, but purely reflecting its sample-point behavior only. At first glance, it may appear that the discrete-time design performs better. But when we actually compute the lifted sampled-data frequency response in the sense defined in Definition 1, it becomes obvious that the sampled-data design is far superior. The dashed curve shows the frequency response of the closed-loop, i.e., that of $G(s)$ connected with the discrete-time designed K_d. The response is similar to the discrete-time frequency response in low frequency, but exhibits a very sharp peak around the Nyquist frequency (i.e., half the sampling frequency; in the present case, $\pi/h \sim 6.28$ rad/s, i.e., $1/2h = 1$ Hz).

This can also be verified from the initial-state responses Fig. 7 with $x(0) = (1, 1)$. The solid curve shows the sampled-data design and the dashed curve the discrete-time one. Both responses decay to zero rapidly *at sampled instants* as shown by the circles for the discrete-time design. But the discrete-time design exhibits very large ripples, with period approximately 1 s. This corresponds to 1 Hz, which is the same as $2\pi = \pi/h$ [rad/s], i.e., the Nyquist frequency. This is precisely captured in the lifted frequency response in Fig. 6.

It is worth noting that when we take the sampling period h smaller, the response for the discrete-time design becomes even more oscillatory and shows a very high peak in the frequency response.
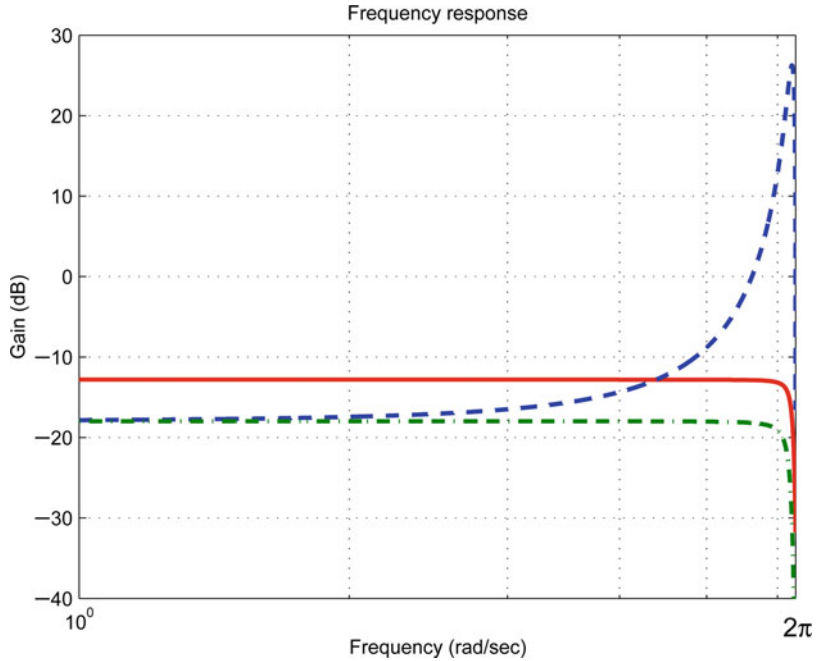
**Optimal Sampled-Data
Control, Fig. 5**
Disturbance rejection
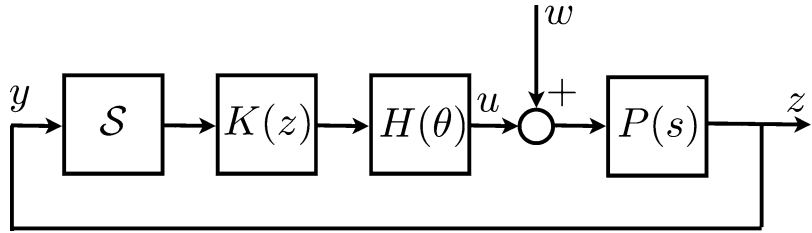
**Optimal Sampled-Data
Control, Fig. 6** Frequency
responses $h = 0.5$

Summary, Bibliographical Notes, and Future Directions

We have given a short summary of the main achievements of modern sampled-data control theory. Particularly, we have reviewed how the technique of lifting resolved the intrinsic difficulty arising from the mixture of two distinct time sets: continuous and discrete. This idea further led to the new notions of transfer operators and frequency response. These notions together enabled us to treat optimal sampled-data control problems in a unified and transparent way. We have outlined how the sampled-data H^∞ control problem can equivalently be reduced to a corresponding discrete-time H^∞ problem, without sacrificing the performance in the intersample behavior. This has been exemplified by a numerical example.

There are other performance indices for optimality, typically those arising from H^2 and L^1 norms. These problems have also been studied extensively, and fairly complete solutions are available. For the lack of space, we cannot list all references, and the reader is referred to Chen and Francis (1995) and Yamamoto (1999) for a more concrete survey and references therein.

For classical treatments of sampled-data control, it is instructive to consult Jury (1958) and Ragazzini and Franklin (1958). The textbook Åström and Wittenmark (1996) covers both classical and modern aspects of digital control. For a mathematical background of the computation of adjoints treated in section "H^∞ Norm Computation and Reduction to Finite Dimension," consult Yamamoto (2012) as well as Yamamoto (1993).

Since control devices are now mostly digital, the importance of sampled-data control will
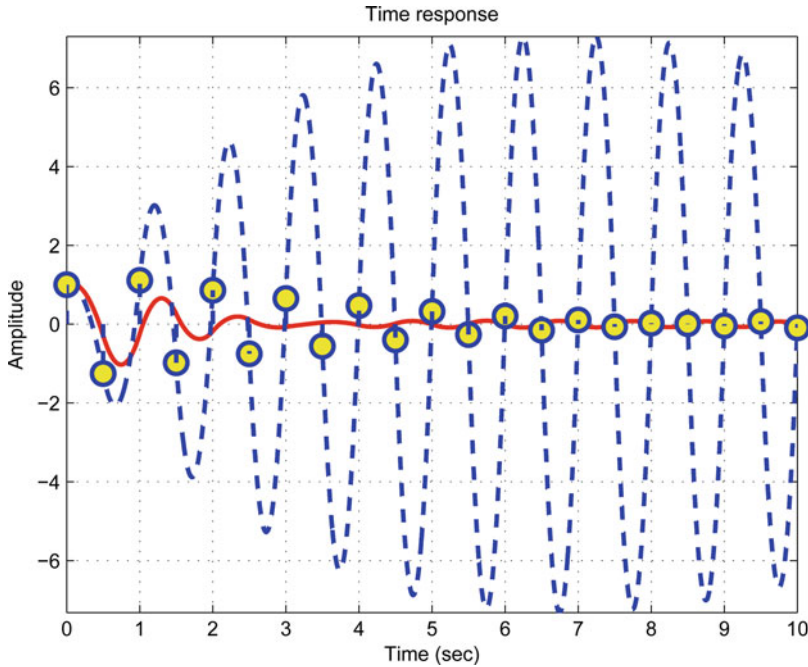
Optimal Sampled-Data Control, Fig. 7 Initial-state responses $h = 0.5$

definitely increase. While the linear, time-invariant case as treated here is now fairly complete, sampled-data control for a nonlinear or an infinite-dimensional plant seems to be still quite an open issue, although it is unclear if the methodology treated here is effective for such classes of plants.

Sampled-data control has much to do with signal processing. Indeed, since it can optimize continuous-time performance, it can shed a new light on digital signal processing. Traditionally, Shannon's paradigm based on the perfect band-limiting hypothesis and the sampling theorem has been prevalent in the signal processing community. Since the sampling theorem opts for perfect reconstruction, the resulting theory reduces mostly to discrete-time problems. In other words, the intersample information is buried in the sampling theorem. It should, however, be noted that the very stringent band-limiting hypothesis is almost never satisfied in reality, and various approximations are necessitated. In contrast, sampled-data control can provide an optimal platform for dealing with and optimizing the response between sampling

points when the band-limiting hypothesis does not hold. See, for example, Yamamoto et al. (2012) and Nagahara and Yamamoto (2012) for the idea and some efforts in this direction.

Cross-References

▶ Control Applications in Audio Reproduction
▶ H_2 Optimal Control
▶ H-Infinity Control
▶ Optimal Control via Factorization and Model Matching

Bibliography

Åström KJ, Wittenmark B (1996) Computer controlled systems—theory and design, 3rd edn. Prentice Hall, Upper Saddle River

Bamieh B, Pearson JB (1992) A general framework for linear periodic systems with applications to H_∞ sampled-data control. IEEE Trans Autom Control 37:418–435

Bamieh B, Pearson JB, Francis BA, Tannenbaum A (1991) A lifting technique for linear periodic systems with applications to sampled-data control systems. Syst Control Lett 17:79–88

Cantoni M, Glover K (1997) H_∞ sampled-data synthesis and related numerical issues. Automatica 33:2233–2241

Chen T, Francis BA (1990) On the $\mathcal{L}_2$-induced norm of a sampled-data system. Syst Control Lett 15:211–219

Chen T, Francis BA (1995) Optimal sampled-data control systems. Springer, New York

Jury EI (1958) Sampled-data control systems. Wiley, New York

Kabamba PT, Hara S (1993) Worst case analysis and design of sampled data control systems. IEEE Trans Autom Control 38:1337–1357

Nagahara M, Yamamoto, Y (2012) Frequency domain min-max optimization of noise-shaping Delta-Sigma modulators. IEEE Trans Signal Process 60:2828–2839

Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York

Sivashankar N, Khargonekar PP (1994) Characterization and computation of the $\mathcal{L}_2$-induced norm of sampled-data systems. SIAM J Control Optim 32:1128–1150

Tadmor G (1991) Optimal $\mathcal{H}_\infty$ sampled-data control in continuous time systems. In: Proceedings of ACC'91, Boston, Massachusetts, pp 1658–1663

Toivonen HT (1992) Sampled-data control of continuous-time systems with an $\mathcal{H}_\infty$ optimality criterion. Automatica 28:45–54

Yamamoto Y (1990) New approach to sampled-data systems: a function space method. In: Proceedings of 29th CDC, Honolulu, Hawaii, pp 1882–1887

Yamamoto Y (1993) On the state space and frequency domain characterization of H^∞-norm of sampled-data systems. Syst Control Lett 21:163–172

Yamamoto Y (1994) A function space approach to sampled-data control systems and tracking problems. IEEE Trans Autom Control 39:703–712

Yamamoto Y (1999) Digital control. In: Webster JG (ed) Wiley encyclopedia of electrical and electronics engineering, vol 5. Wiley, New York, pp 445–457

Yamamoto Y (2012) From vector spaces to function spaces—introduction to functional analysis with applications. SIAM, Philadelphia

Yamamoto Y, Khargonekar PP (1996) Frequency response of sampled-data systems. IEEE Trans Autom Control 41:166–176

Yamamoto Y, Nagahara M, Khargonekar PP (2012) Signal reconstruction via H^∞ sampled-data control theory—Beyond the Shannon paradigm. IEEE Trans Signal Process 60:613–625

Optimization Algorithms for Model Predictive Control

Moritz Diehl
Department of Microsystems Engineering (IMTEK), University of Freiburg, Freiburg, Germany
ESAT-STADIUS/OPTEC, KU Leuven, Leuven-Heverlee, Belgium

Abstract

This entry reviews optimization algorithms for both linear and nonlinear model predictive control (MPC). Linear MPC typically leads to specially structured convex quadratic programs (QP) that can be solved by structure exploiting active set, interior point, or gradient methods. Nonlinear MPC leads to specially structured nonlinear programs (NLP) that can be solved by sequential quadratic programming (SQP) or nonlinear interior point methods.

Keywords

Banded matrix factorization; Convex optimization; Karush-Kuhn-Tucker (KKT) conditions; Sparsity exploitation

Introduction

Model predictive control (MPC) needs to solve at each sampling instant an optimal control problem with the current system state $\bar{x}_0$ as initial value. MPC optimization is almost exclusively based on the so-called *direct approach* which first discretizes the continuous time system to obtain a discrete time optimal control problem (OCP). This OCP has as optimization variables a state trajectory $X = [x_0^\top, \ldots, x_N^\top]^\top$ with $x_i \in \mathbb{R}^{n_x}$ for $i = 0, \ldots, N$ and a control trajectory $U = [u_0^\top, \ldots, u_{N-1}^\top]^\top$ with $u_i \in \mathbb{R}^{n_u}$ for $i = 0, \ldots, N - 1$. For simplicity of presentation, we

restrict ourselves to the time-independent case, and the OCP we treat in this article is stated as follows:

$$\underset{X, U}{\text{minimize}} \quad \sum_{i=0}^{N-1} L(x_i, u_i) \; + \; E\left(x_N\right) \qquad \text{(1a)}$$

$$\text{subject to} \quad x_0 - \bar{x}_0 = 0, \qquad\qquad\qquad \text{(1b)}$$

$$x_{i+1} - f(x_i, u_i) = 0, \quad i = 0, \ldots, N-1, \qquad \text{(1c)}$$

$$h(x_i, u_i) \leq 0, \quad i = 0, \ldots, N-1, \qquad \text{(1d)}$$

$$r\left(x_N\right) \leq 0. \qquad\qquad\qquad \text{(1e)}$$

The MPC objective is stated in Eq. (1a), the system dynamics enter via Eq. (1c), while path and terminal constraints enter via Eqs. (1d) and (1e). All functions are assumed to be differentiable and to have appropriate dimensions ($h(x, u) \in \mathbb{R}^{n_h}$ and $r(x) \in \mathbb{R}^{n_r}$). Note that $\bar{x}_0 \in \mathbb{R}^{n_x}$ is not an optimization variable, but a parameter upon which the OCP depends via the initial value constraint in Eq. (1b). The optimal solution trajectories depend only on this value and can thus be denoted by $X^*(\bar{x}_0)$ and $U^*(\bar{x}_0)$. Obtaining them, in particular the first control value $u_0^*(\bar{x}_0)$, as fast and reliably as possible for each new value of $\bar{x}_0$ is the aim of all MPC optimization algorithms. The most important dividing line is between convex and non-convex optimal control problems (OCP). If the OCP is convex, algorithms exist that find a global solution reliably and in computable time. If the OCP is not convex, one usually needs to be satisfied with approximations of locally optimal solutions. The OCP (1) is convex if the objective (1a) and all components of the inequality constraint functions (1d) and (1e) are convex and if the equality constraints (1c) are linear.

We typically speak of *linear MPC* when the OCP to be solved is convex, and otherwise of *nonlinear MPC*.

General Algorithmic Features for MPC Optimization

In MPC we would dream to have the solution to a new optimal control problem instantly, which is impossible due to computational delays. Several ideas can help us to deal with this issue.

Off-line Precomputations and Code Generation

As consecutive MPC problems are similar and differ only in the value $\bar{x}_0$, many computations can be done once and for all before the MPC controller execution starts. Careful preprocessing and code optimization for the model routines is essential, and many tools automatically generate custom solvers in low-level languages. The generated code has fixed matrix and vector dimensions, has no online memory allocations, and contains a minimal number of if-then-else statements to ensure a smooth computational flow.

Delay Compensation by Prediction

When we know how long our computations for solving an MPC problem will take, it is a good idea *not* to address a problem starting at the current state but to simulate at which state the system will be when we will have solved the problem. This can be done using the MPC system model and the open-loop control inputs that we will apply in the meantime. This feature is used in many practical MPC schemes with non-negligible computation time.

Division into Preparation and Feedback Phase

A third ingredient of several MPC algorithms is to divide the computations in each sampling time into a preparation phase and a feedback phase. The more CPU intensive *preparation phase* is performed with a predicted state $\bar{x}_0$, before the most current state estimate, say $\bar{x}_0'$, is available. Once $\bar{x}_0'$ is available, the *feedback phase* delivers quickly an *approximate* solution to the optimization problem for $\bar{x}_0'$.

Warmstarting and Shift

An obvious way to transfer solution information from one solved MPC problem to the next one uses the existing optimal solution as an initial guess to start the iterative solution procedure of the next problem. We can either directly use the existing solution without modification for warmstarting or we can first shift it in order to

account for the advancement of time, which is particularly advantageous for systems with time-varying dynamics or objectives.

Iterating While the Problem Changes

A last important ingredient of some MPC algorithms is the idea to work on the optimization problem while it changes, i.e., to never iterate the optimization procedure to convergence for an MPC problem getting older and older during the iterations but to rather work with the most current information in each new iteration.

Convex Optimization for Linear MPC

Linear MPC is based on a linear system model of the form $x_{i+1} = Ax_i + Bu_i$ and convex objective and constraint functions in (1a), (1d), and (1e). The most widespread linear MPC setting uses a convex quadratic objective function and affine constraints and solves the following quadratic program (QP):

$$\underset{X, U}{\text{minimize}} \quad \frac{1}{2} \sum_{i=0}^{N-1} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + \frac{1}{2} x_N^\top P x_N \tag{2a}$$

subject to
$$x_0 - \bar{x}_0 = 0, \tag{2b}$$

$$x_{i+1} - Ax_i - Bu_i = 0, i = 0, \dots, N-1, \tag{2c}$$

$$b + Cx_i + Du_i \leq 0, i = 0, \dots, N-1, \tag{2d}$$

$$c + Fx_N \leq 0. \tag{2e}$$

Here, b, c are vectors and Q, S, R, P, C, D, F matrices, and matrices $\begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix}$ and P are symmetric and positive semi-definite to ensure the QP is convex.

Sparsity Exploitation

The QP (2) has a specific sparsity structure that can be exploited in different ways. One way is to reduce the variable space by a procedure called *condensing* and then to solve a smaller-scale QP

instead of (2). Another way is to use a *banded matrix factorization*.

Condensing

The constraints (2b) and (2c) can be used to eliminate the state trajectory X. This yields an equivalent but smaller-scale QP of the following form:

$$\underset{U \in \mathbb{R}^{Nn_u}}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix}^\top \begin{bmatrix} H & G \\ G^\top & J \end{bmatrix} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix} \tag{3a}$$

subject to $\quad d + K\bar{x}_0 + MU \leq 0. \tag{3b}$

The number of inequality constraints is the same as in the original QP (2) and given by $m = Nn_h + n_r$. Note that in the simplest case without inequalities ($m = 0$), the solution $U^*(\bar{x}_0)$ of the condensed QP can be obtained by setting the gradient of the objective to zero, i.e., by solving $HU^*(\bar{x}_0) + G\bar{x}_0 = 0$. The factorization of a dense matrix H with dimension $Nn_u \times Nn_u$ needs $O(N^3 n_u^3)$ arithmetic operations, i.e., the computational cost of condensing-based algorithms typically grows cubically with the horizon length N.

Banded Matrix Factorization

An alternative way to deal with the sparsity is best sketched at hand of a sparse convex QP (2) without inequality constraints (2d) and (2e). We define the vector of Lagrange multipliers $Y = [y_0^\top, \dots, y_N^\top]^\top$ and the Lagrangian function by

$$\mathcal{L}(X, U, Y) = y_0^\top (x_0 - \bar{x}_0) + \frac{1}{2} x_N^\top P x_N$$
$$+ \frac{1}{2} \sum_{i=0}^{N-1} \begin{bmatrix} x_i \\ u_i \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} x_i \\ u_i \end{bmatrix} + y_{i+1}^\top$$
$$(x_{i+1} - Ax_i + Bu_i). \tag{4}$$

If we reorder all unknowns that enter the Lagrangian and summarize them in the vector

$$W = [y_0^\top, x_0^\top, u_0^\top, y_1^\top, x_1^\top, u_1^\top, \dots, y_N^\top, x_N^\top]^\top$$

the optimal solution $W^*(\bar{x}_0)$ is uniquely characterized by the first-order optimality condition

O

$$\nabla_W \mathcal{L}(W^*) = 0.$$

Due to the linear quadratic dependence of $\mathcal{L}$ on W, this is a block-banded linear equation in the unknown W^*:

$$\begin{bmatrix} 0 & I & & & & & & \\ I & Q & S & -A^\top & & & & \\ & S^\top & R & -B^\top & & & & \\ & -A & -B & 0 & I & & & \\ & & & I & Q & \cdot & \cdot & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & 0 & I \\ & & & & & & I & P \end{bmatrix} W^* = \begin{bmatrix} \bar{x}_0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}.$$

Because the above matrix has nonzero elements only on a band of width $2n_x + n_u$ around the diagonal, it can be factorized with a computational cost of order $O(N(2n_x + n_u)^3)$.

Treatment of Inequalities

An important observation is that both the uncondensed QP (2) and the equivalent condensed QP (3) typically fall into the class of strictly convex parametric quadratic programs: the solution $U^*(\bar{x}_0)$, $X^*(\bar{x}_0)$ is unique and depends piecewise affinely and continuously on the parameter $\bar{x}_0$. Each affine piece of the solution map corresponds to one *active set* and is valid on one polyhedral *critical region* in parameter space. This observation forms the basis of *explicit MPC algorithms* which precompute the map $u_0^*(\bar{x}_0)$, but it can also be exploited in online algorithms for quadratic programming, which are the focus of this section.

We sketch the different ways of how to treat inequalities only for the condensed QP (3), but they can equally be applied in sparse QP algorithms that directly address (2). The optimal solution $U^*(\bar{x}_0)$ for a strictly convex QP (3) is – together with the corresponding vector of Lagrange multipliers, or dual variables $\lambda^*(\bar{x}_0) \in \mathbb{R}^m$ – uniquely characterized by the so-called Karush-Kuhn-Tucker (KKT) conditions, which we omit for brevity. There are three big families of solution algorithms for inequality-constrained QPs that differ in the way they treat inequalities: *active set methods*, *interior point methods*, and *gradient projection methods*.

Active Set Methods

The optimal solution of the QP is characterized by its *active set*, i.e., the set of inequality constraints (3b) that are satisfied with equality at this point. If one would know the active set for a given problem instance $\bar{x}_0$, it would be easy to find the solution. Active set methods work with guesses of the active set which they iteratively refine. In each iteration, they solve one linear system corresponding to a given guess of the active set. If the KKT conditions are satisfied, the optimal solution is found; if they are not, another division into active and inactive constraints needs to be tried. A crucial observation is that an existing factorization of the linear system can be reused to a large extent when only one constraint is added or removed from the guess of the active set. Many different active set strategies exist, three of which we mention: *Primal active set methods* first find a feasible point and then add or remove active constraints, always keeping the primal variables U feasible. Due to the difficulty of finding a feasible point first, they are difficult to warmstart in the context of MPC optimization. *Dual active set strategies* always keep the dual variables λ positive. They can easily be warmstarted in the context of MPC. *Parametric* or *online active set strategies* ensure that all iterates stay primal and dual feasible and go on a straight line in parameter space from a solved QP problem to the current one, only updating the active set when crossing the boundary between critical regions, as implemented in the online QP solver qpOASES.

Active set methods are very competitive in practice, but their worst case complexity is not polynomial. They are often used together with the condensed QP formulation, for which each active set change is relatively cheap. This is particularly advantageous if an existing factorization can be kept between subsequent MPC optimization problems.

Interior Point Methods

Another approach is to replace the KKT conditions by a smooth approximation that uses a small positive parameter $\tau > 0$:

$$HU^* + G\bar{x}_0 + M^\top \lambda^* = 0, \tag{5a}$$

$$\left(d + K\bar{x}_0 + MU^*\right)_i \lambda_i^* + \tau = 0, \ i = 1, \ldots, m.$$
(5b)

These conditions form a smooth nonlinear system of equations that uniquely determines a primal dual solution $U^*(\bar{x}_0, \tau)$ and $\lambda^*(\bar{x}_0, \tau)$ in the interior of the feasible set. They are not equivalent to the KKT conditions, but for $\tau \to 0$, their solution tends to the exact QP solution. An interior point algorithm solves the system (5a) and (5b) by Newton's method. Simultaneously, the path parameter τ, that was initially set to a large value, is iteratively reduced, making the nonlinear set of equations a closer approximation of the original KKT system. In each Newton iteration, a linear system needs to be factored and solved, which constitutes the major computational cost of an interior point algorithm. For the condensed QP (3) with dense matrices H, M, the cost per Newton iteration is of order $O(N^3)$. But the interior point algorithm can also be applied to the uncondensed sparse QP (2), in which case each iteration has a runtime of order $O(N)$. In practice, for both cases, 10–30 Newton iterations usually suffice to obtain very accurate solutions. As an interior point method needs always to start with a high value of τ and then reduces it during the iterations, warmstarting is of minor benefit. There exist efficient code generation tools that export convex interior point solvers as plain C-code such as CVXGEN and FORCES.

Gradient Projection Methods
Gradient projection methods do not need to factorize any matrix but only evaluate the gradient of the objective function $HU^{[k]} + G\bar{x}_0$ in each iteration. They can only be implemented efficiently if the feasible set is a simple set in the sense that a projection $\mathcal{P}(U)$ on this set is very cheap to compute, as, e.g., for upper and lower bounds on the variables U, and if we know an upper bound $L_H > 0$ on the eigenvalues of the Hessian H. The simple gradient projection algorithm starts with an initialization $U^{[0]}$ and proceeds as follows:

$$U^{[k+1]} = \mathcal{P}\left(U^{[k]} - \frac{1}{L_H}(HU^{[k]} + G\bar{x}_0)\right).$$

An improved version of the gradient projection algorithm is called the *optimal* or *fast gradient method* and has probably the best possible iteration complexity of all gradient type methods. All variants of gradient projection algorithms are easy to warmstart. Though they are not as versatile as active set or interior point methods, they have short code sizes and can offer advantages on embedded computational hardware, such as the code generated by the tool FIOR-DOS.

Optimization Algorithms for Nonlinear MPC

When the dynamic system $x_{i+1} = f(x_i, u_i)$ is not affine, the optimal control problem (1) is non-convex, and we speak of a nonlinear MPC (NMPC) problem. NMPC optimization algorithms only aim at finding a locally optimal solution of this problem, and they usually do it in a Newton-type framework. For ease of notation, we summarize problem (1) in the form of a general nonlinear programming problem (NLP):

$$\begin{aligned}
\underset{X, U}{\text{minimize}} \quad & \Phi(X, U) && \text{(6a)} \\
\text{subject to} \quad & G_{\text{eq}}(X, U, \bar{x}_0) = 0, && \text{(6b)} \\
& G_{\text{ineq}}(X, U) \leq 0. && \text{(6c)}
\end{aligned}$$

Let us first discuss a fundamental choice that regards the problem formulation and number of optimization variables.

Simultaneous vs. Sequential Formulation
When an optimization algorithm addresses problem (6) iteratively, it works intermediately with nonphysical, infeasible trajectories that violate the system constraints (6b). Only at the optimal solution the constraint residual is brought to zero and a physical simulation is achieved. We speak

of a *simultaneous approach* to optimal control because the algorithm solves the simulation and the optimization problems simultaneously. Variants of this approach are *direct discretization* or *direct multiple shooting*.

On the other hand, the equality constraints (6b) could easily be eliminated by a nonlinear forward simulation for given initial value $\bar{x}_0$ and control trajectory U, similar to condensing in linear MPC. Such a forward simulation generates a state trajectory $X_{\text{sim}}(\bar{x}_0, U)$ such that for the given value of $\bar{x}_0, U$ the equality constraints (6b) are automatically satisfied: $G_{\text{eq}}(X_{\text{sim}}(\bar{x}_0, U), U, \bar{x}_0) = 0$. Inserting this map into the NLP (6) allows us to formulate an equivalent optimization problem with a reduced variable space:

$$\underset{U}{\text{minimize}} \qquad \Phi(X_{\text{sim}}(\bar{x}_0, U), U) \qquad (7a)$$

$$\text{subject to} \quad G_{\text{ineq}}(X_{\text{sim}}(\bar{x}_0, U), U) \leq 0. \quad (7b)$$

When solving this reduced problem with an iterative optimization algorithm, we sequentially simulate and optimize the system, and we speak of the *sequential approach* to optimal control.

The sequential approach has a lower dimensional variable space and is thus easier to use with a black-box NLP solver. On the other hand, the simultaneous approach leads to a sparse NLP and is better able to deal with unstable nonlinear systems. In the remainder of this section, we thus only discuss the specific structure of the simultaneous approach.

Newton-Type Optimization

In order to simplify notation further, we summarize and reorder all optimization variables U and X in a vector $V = [x_0^\top, u_0^\top, \ldots, x_{N-1}^\top, u_{N-1}^\top, x_N^\top]^\top$ and use the same problem function names as in (6) also with the new argument V.

As in the section on linear MPC, we can introduce multipliers Y for the equalities and λ for the inequalities and define the Lagrangian

$$\mathcal{L}(V, Y, \lambda) = \Phi(V) + Y^\top G_{\text{eq}}(V, \bar{x}_0)$$

$$+ \lambda^\top G_{\text{ineq}}(V). \qquad (8)$$

All Newton-type optimization methods try to find a point satisfying the KKT conditions by using successive linearizations of the problem functions and Lagrangian. For this aim, starting with an initial guess $(V^{[0]}, Y^{[0]}, \lambda^{[0]})$, they generate sequences of primal-dual iterates $(V^{[k]}, Y^{[k]}, \lambda^{[k]})$.

An observation that is crucial for the efficiency of all NMPC algorithms is that the Hessian of the Lagrangian is at a current iterate given by a matrix of the form

$$\nabla_V^2 \mathcal{L}(\cdot) = \begin{bmatrix} Q_0^{[k]} & S_0^{[k]} & & & & \\ S_0^{[k],\top} & R_0^{[k]} & & & & \\ & & Q_1^{[k]} & S_1^{[k]} & & \\ & & S_1^{[k],\top} & R_1^{[k]} & & \\ & & & & \ddots & \\ & & & & & \ddots & \\ & & & & & & P^{[k]} \end{bmatrix}.$$

This block sparse matrix structure makes it possible to use in each Newton-type iteration the same sparsity-exploiting linear algebra techniques as outlined in section "Convex Optimization for Linear MPC" for the linear MPC problem.

Major differences exist on how to treat the inequality constraints, and the two big families of Newton-type optimization methods are *sequential quadratic programming (SQP)* methods and *nonlinear interior point (NIP)* methods.

Sequential Quadratic Programming (SQP)

A first variant to iteratively solve the KKT system is to linearize all nonlinear functions at the current iterate $(V^{[k]}, Y^{[k]}, \lambda^{[k]})$ and to find a new solution guess from the solution of a quadratic program (QP):

$$\underset{V}{\text{minimize}} \qquad \Phi_{\text{quad}}(V; V^{[k]}, Y^{[k]}, \lambda^{[k]}) \qquad (9a)$$

$$\text{subject to} \qquad G_{\text{eq,lin}}(V, \bar{x}_0; V^{[k]}) = 0, \qquad (9b)$$

$$G_{\text{ineq,lin}}(V; V^{[k]}) \leq 0. \qquad (9c)$$

Here, the subindex "lin" in the constraints $G_{\cdot,\text{lin}}$ expresses that a first-order Taylor expansion at

$V^{[k]}$ is used, while the QP objective is given by $\Phi_{\text{quad}}(V; V^{[k]}, Y^{[k]}, \lambda^{[k]}) = \Phi_{\text{lin}}(V; V^{[k]}) + \frac{1}{2}(V - V^{[k]})^\top \nabla_V^2 \mathcal{L}(\cdot)(V - V^{[k]})$. Note that the QP has the same sparsity structure as the QP (2) resulting from linear MPC, with the only difference that all matrices are now time varying over the MPC horizon. In the case that the Hessian matrix is positive semi-definite, this QP is convex so that global solutions can be found reliably with any of the methods from section "Convex Optimization for Linear MPC." The solution of the QP along with the corresponding constraint multipliers gives the next SQP iterate $(V^{[k+1]}, Y^{[k+1]}, \lambda^{[k+1]})$. Apart from the presented "exact Hessian" SQP variant, which has quadratic convergence speed, several other SQP variants exist, which make use of other Hessian approximations. A particularly useful Hessian approximation for NMPC is possible if the original objective function $\Phi(V)$ is convex quadratic, and the resulting SQP variant is called the *generalized Gauss-Newton* method. In this case, one can just use the original objective as cost function in the QP (9a), resulting in convex QP subproblems and (often fast) linear convergence speed.

Nonlinear Interior Point (NIP) Method

In contrast to SQP methods, an alternative way to address the solution of the KKT system is to replace the last nonsmooth KKT conditions by a smooth nonlinear approximation, with $\tau > 0$:

$$\nabla_V \mathcal{L}(V^*, Y^*, \lambda^*) = 0 \qquad (10a)$$

$$G_{\text{eq}}(V^*, \bar{x}_0) = 0 \qquad (10b)$$

$$G_{\text{ineq},i}(V^*)\,\lambda_i^* + \tau = 0, \quad i = 1, \dots, m. \qquad (10c)$$

We summarize all variables in a vector $W = [V^\top, Y^\top, \lambda^\top]^\top$ and summarize the above set of equations as

$$G_{\text{NIP}}(W, \bar{x}_0, \tau) = 0. \qquad (11)$$

The resulting root finding problem is then solved with Newton's method, for a descending sequence of path parameters $\tau^{[k]}$. The NIP method proceeds thus exactly as in an interior point method for convex problems, with the only difference that it has to re-linearize all problem functions in each iteration. An excellent software implementation of the NIP method is given in the form of the code IPOPT.

Continuation Methods and Tangential Predictors

In nonlinear MPC, a sequence of OCPs with different initial values $\bar{x}_0^{[0]}, \bar{x}_0^{[1]}, \bar{x}_0^{[2]}, \dots$ is solved. For the transition from one problem to the next, it is beneficial to take into account the fact that the optimal solution $W^*(\bar{x}_0)$ depends almost everywhere differentiably on $\bar{x}_0$. The concept of a continuation method is most easily explained in the context of an NIP method with fixed path parameter $\bar{\tau} > 0$. In this case, the solution $W^*(\bar{x}_0, \bar{\tau})$ of the smooth root finding problem $G_{\text{NIP}}(W^*(\bar{x}_0, \bar{\tau}), \bar{x}_0, \bar{\tau}) = 0$ from Eq. (11) is smooth with respect to $\bar{x}_0$. This smoothness can be exploited by making use of a *tangential predictor* in the transition from one value of $\bar{x}_0$ to another. Unfortunately, the interior point solution manifold is strongly nonlinear at points where the active set changes, and the tangential predictor is not a good approximation when we linearize at such points.

Generalized Tangential Predictor and Real-Time Iterations

In fact, the true NLP solution is not determined by a smooth root finding problem (10a)–(3) but by the (nonsmooth) KKT conditions. The solution manifold has smooth parts when the active set does not change, but non-differentiable points occur whenever the active set changes. We can deal with this fact naturally in an SQP framework by solving one QP of form (9) in order to generate a tangential predictor that is also valid in the presence of active set changes. In the extreme case that only one such QP is solved per sampling time, we speak of a *real-time iteration (RTI)* algorithm. The computations in each iteration can be subdivided into two phases, the *preparation phase*, in which the derivatives are computed and the QP is condensed, and the *feedback phase*, which

only starts once $\bar{x}_0^{[k+1]}$ becomes available and in which only a condensed QP of form (3) is solved, minimizing the feedback delay. This NMPC algorithm can be generated as plain C-code, e.g., by the tool ACADO. Another class of real-time NMPC algorithms based on a continuation method can be generated by the tool AutoGenU.

Cross-References

Recommended Reading

Many of the algorithmic ideas presented in this article can be used in different combinations than those presented, and several other ideas had to be omitted for the sake of brevity. Some more details can be found in the following two overview articles on MPC optimization: Binder et al. (2001) and Diehl et al. (2009). The general field of numerical optimal control is treated in Bryson and Ho (1975), Betts (2010), and the even broader field of numerical optimization is covered in the excellent textbooks (Fletcher 1987; Wright 1997; Nesterov 2004; Gill et al. 1999; Nocedal and Wright 2006; Biegler 2010). General purpose open-source software for MPC and NMPC is described in the following papers: FORCES (Domahidi et al. 2012), CVXGEN (Mattingley and Boyd 2009), qpOASES (Ferreau et al. 2008), FiOrdOs (Richter et al. 2011), AutoGenU (Ohtsuka and Kodama 2002), ACADO (Houska et al. 2011), and IPOPT (Wächter and Biegler 2006).

Bibliography

Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia

Biegler LT (2010) Nonlinear programming. SIAM, Philadelphia

Binder T, Blank L, Bock HG, Bulirsch R, Dahmen W, Diehl M, Kronseder T, Marquardt W, Schlöder JP, Stryk OV (2001) Introduction to model based optimization of chemical processes on moving horizons. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems: state of the art. Springer, Berlin, pp 295–340

Bryson AE, Ho Y-C (1975) Applied optimal control. Wiley, New York

Diehl M, Ferreau HJ, Haverbeke N (2009) Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Nonlinear model predictive control. Lecture notes in control and information sciences, vol 384. Springer, Berlin, pp 391–417

Domahidi A, Zgraggen A, Zeilinger MN, Morari M, Jones CN (2012) Efficient interior point methods for multistage problems arising in receding horizon control. In: IEEE conference on decision and control (CDC), Maui, Dec 2012, pp 668–674

Ferreau HJ, Bock HG, Diehl M (2008) An online active set strategy to overcome the limitations of explicit MPC. Int J Robust Nonlinear Control 18(8): 816–830

Fletcher R (1987) Practical methods of optimization, 2nd edn. Wiley, Chichester

Gill PE, Murray W, Wright MH (1999) Practical optimization. Academic, London

Houska B, Ferreau HJ, Diehl M (2011) An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. Automatica 47(10): 2279–2285

Mattingley J, Boyd S (2009) Automatic code generation for real-time convex optimization. In: Convex optimization in signal processing and communications. Cambridge University Press, New York, pp 1–43

Nesterov Y (2004) Introductory lectures on convex optimization: a basic course. Applied optimization, vol 87. Kluwer, Boston

Nocedal J, Wright SJ (2006) Numerical optimization. Springer series in operations research and financial engineering, 2nd edn. Springer, New York

Ohtsuka T, Kodama A (2002) Automatic code generation system for nonlinear receding horizon control. Trans Soc Instrum Control Eng 38(7): 617–623

Richter S, Morari M, Jones CN (2011) Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method. In: 50th IEEE conference on decision and control and European control conference (CDC-ECC), Orlando, Dec 2011, pp 5223–5229

Wächter A, Biegler LT (2006) On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math Program 106(1):25–57

Wright SJ (1997) Primal-dual interior-point methods. SIAM, Philadelphia

Optimization Based Robust Control

Didier Henrion
LAAS-CNRS, University of Toulouse, Toulouse,
France
Faculty of Electrical Engineering, Czech
Technical University in Prague, Prague,
Czech Republic

Abstract

This entry describes the basic setup of linear robust control and the difficulties typically encountered when designing optimization algorithms to cope with robust stability and performance specifications.

Keywords

Linear systems; Optimization; Robust control

Linear Robust Control

Robust control allows dealing with uncertainty affecting a dynamical system and its environment. In this section, we assume that we have a mathematical model of the dynamical system without uncertainty (the so-called nominal system) jointly with a mathematical model of the uncertainty. We restrict ourselves to **linear systems**: if the dynamical system we want to control has some nonlinear components (e.g., input saturation), they must be embedded in the uncertainty model. Similarly, we assume that the control system is relatively small scale (low number of states): higher-order dynamics (e.g., highly oscillatory but low energy components) are embedded in the uncertainty model. Finally, for conciseness, we focus exclusively on continuous-time systems, even though most of the techniques described in this section can be transposed readily to discrete-time systems.

Our control system is described by the first-order ordinary differential equation

$$\dot{x} = A(\delta) x + D(\delta) u$$
$$y = C(\delta) x$$

where as usual $x \in \mathbb{R}^n$ denotes the states, $u \in \mathbb{R}^m$ denotes the controlled inputs, and $y \in \mathbb{R}^p$ denotes the measured outputs, all depending on time t, with $\dot{x}$ denoting the time derivative of x. The system is subject to uncertainty and this is reflected by the dependence of matrices A, B, and C on uncertain parameter δ which is typically time varying and restricted to some bounded set

$$\delta \in \Delta \subset \mathbb{R}^q.$$

A linear control law

$$u = Ky$$

modeled by a matrix $K \in \mathbb{R}^{m \times p}$ must be designed to overcome the effect of the uncertainty while optimizing some performance criterion (e.g., pole placement, disturbance rejection, H_2 or H_∞ norm). Sometimes, a relevant performance criterion is that the control should be stabilizing for the largest possible uncertainty (measured, e.g., by some norm on Δ). In this section, for conciseness, we restrict our attention to **static output feedback** control laws, but most of the results can be extended to dynamical output feedback control laws, where the control signal u is the output of a controller (a linear system to be designed) whose input is y.

Uncertainty Models

Amongst the simplest possible uncertainty models, we can find the following:
- **Unstructured uncertainty**, also called norm-bounded uncertainty, where

$$\Delta = \{\delta \in \mathbb{R}^q : ||\delta|| \leq 1\}$$

and the given norm can be a standard vector norm or a more complicated matrix norm if δ is

interpreted as a vector obtained by stacking the column of a matrix

- **Structured uncertainty**, also called polytopic uncertainty, where

$$\Delta = \text{conv} \{\delta_i, i = 1, \ldots, N\}$$

is a polytope modeled as the convex combination of a finite number of given vertices $\delta_i \in \mathbb{R}^q, i = 1, \ldots, N$

We can find more complicated uncertainty models (e.g., combinations of the two above: see Zhou et al. 1996), but to keep the developments elementary, they are not discussed here.

Nonconvex Nonsmooth Robust Optimization

The main difficulties faced when seeking a feedback matrix K are as follows:

- **Nonconvexity**: The stability conditions are typically nonconvex in K.
- **Nondifferentiability**: The performance criterion to be optimized is typically a nondifferentiable function of K.
- **Robustness**: Stability and performance should be ensured for every possible instance of the uncertainty.

So if we are to formulate the robust control problem as an optimization problem, we should be ready to develop and use techniques from nonconvex, nondifferentiable, robust optimization.

Let us first elaborate on the first difficulty faced by optimization-based robust control, namely, the nonconvexity of the stability conditions. In continuous time, stability of a linear system $\dot{x} = Ax$ is equivalent to negativity of the spectral abscissa, which is defined as the maximum real part of the eigenvalues of A:

$$\alpha(A) = \max\{\text{Re } \lambda : \det(\lambda I_n - A) = 0, \ \lambda \in \mathbb{C}\}.$$

It turns out that the open cone of matrices $A \in \mathbb{R}^{n \times n}$ such that $\alpha(A) < 0$ is nonconvex (Ackermann 1993). This is illustrated in Fig. 1 where we represent the set of vectors $K =$

$(k_1, k_2, k_3) \in \mathbb{R}^3$ such that $k_1^2 + k_2^2 + k_3^2 < 1$ and $\alpha(A(K)) < 0$ for

$$A(K) = \begin{pmatrix} -1 & k_1 \\ k_2 & k_3 \end{pmatrix}.$$

There exist various approaches to handling nonconvexity. One possibility consists of building convex inner approximations of the stability region in the parameter space. The approximations can be polytopes, balls, ellipsoids, or more complicated convex objects described by linear matrix inequalities (LMI). The resulting stability conditions are convex, but surely conservative, in the sense that the conditions are only sufficient for stability and not necessary. Another approach to handling nonconvexity consists of formulating the stability conditions algebraically (e.g., via the Routh-Hurwitz stability criterion or its symmetric version by Hermite) and using converging hierarchies of LMI relaxations to solve the resulting nonconvex polynomial optimization problem: see, e.g., Henrion and Lasserre (2004) and Chesi (2010).

The second difficulty characteristic of optimization-based robust control is the potential nondifferentiability of the objective function. Consider for illustration one of the simplest optimization problems which consists of minimizing the spectral abscissa $\alpha(A(K))$ of a matrix $A(K)$ depending linearly on a matrix K. Such a minimization makes sense since negativity of the spectral abscissa is equivalent to system stability. Then typically, $\alpha(A(K))$ is a continuous but non-Lipschitz function of K, which means that its gradient can be unbounded locally. In Fig. 2, we plot the spectral abscissa $\alpha(A(K))$ for

$$A(K) = \begin{pmatrix} 0 & 1 \\ K & -K \end{pmatrix}$$

and $K \in \mathbb{R}$. The function is non-Lipschitz at $K = 0$, at which the global minimum $\alpha(A(0)) = 0$ is achieved. Nonconvexity of the function is also apparent in this example. The lack of convexity and smoothness of the spectral abscissa and other similar performance criteria renders optimization of such functions particularly difficult (Burke

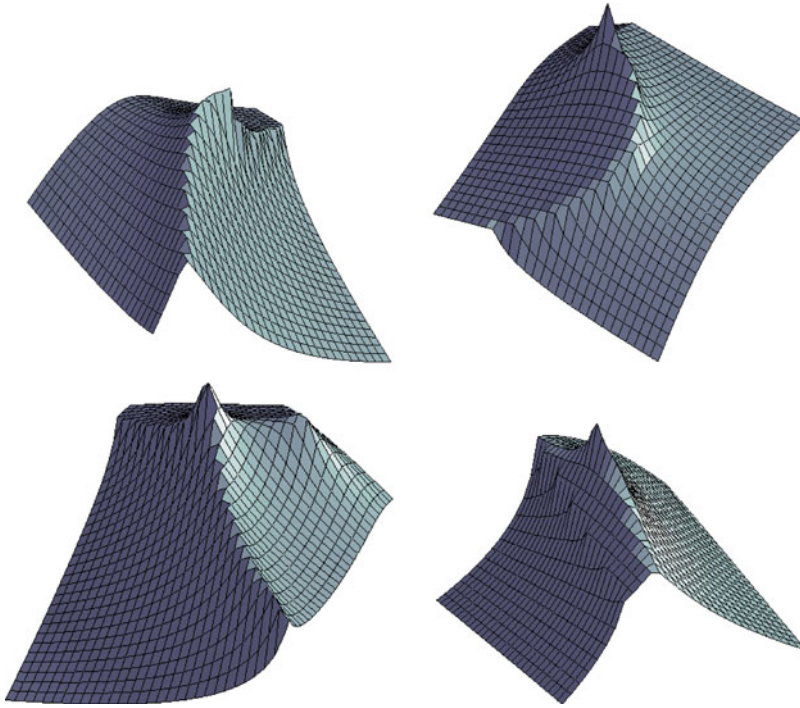**Optimization Based Robust Control, Fig. 1** A nonconvex ball of stable matrices
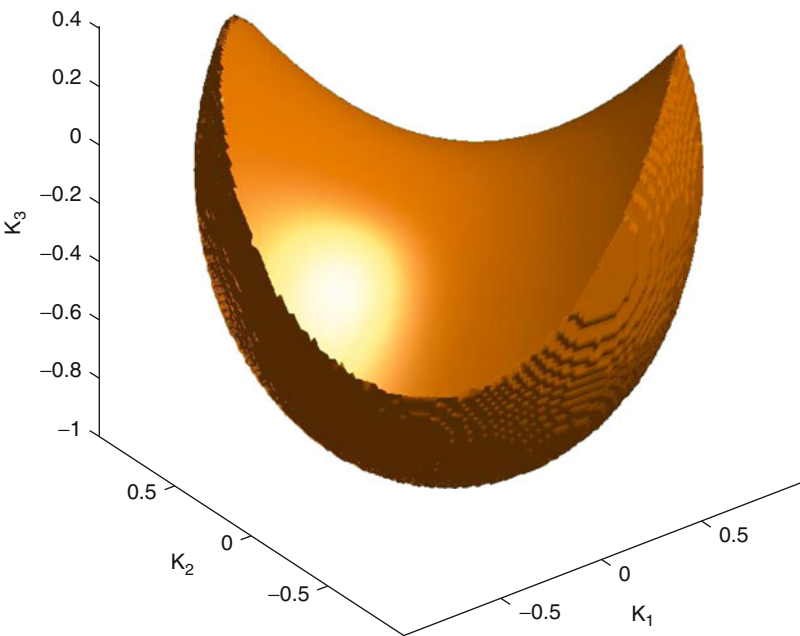
Optimization Based Robust Control, Fig. 2 The spectral abscissa is typically nonconvex and nonsmooth
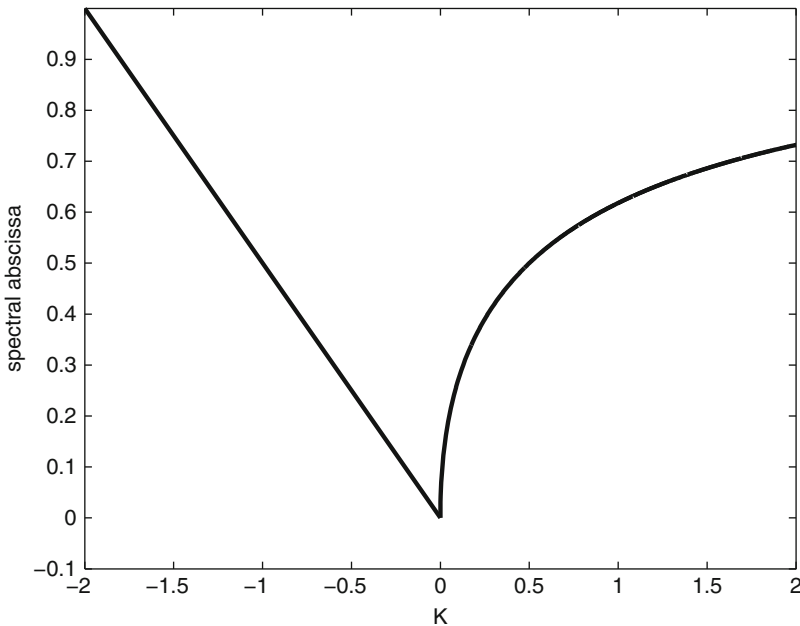
et al. 2001, 2006b). In Fig. 3, we represent graphs of the spectral abscissa (with flipped vertical axis for better visualization) of some small-size matrices depending on two real parameters, with randomly generated parametrization. We observe the typical nonconvexity and lack of smoothness around local and global optima.

The third difficulty for optimization-based robust control is the uncertainty. As explained above, optimization of a performance criterion with respect to controller parameters is already a potentially difficult problem for a nominal system (i.e., when the uncertainty parameter is equal to zero). This becomes even more difficult

Optimization Based Robust Control, Fig. 3 The graph of the negative spectral abscissa for some randomly generated matrix parametrizations

when this optimization must be carried out for all possible instances of the uncertainty δ in Δ. This is where the above assumption that the uncertainty set Δ has a simple description proves useful. If the uncertainty δ is unstructured and not time varying, then it can be handled with the complex stability radius (Ackermann 1993), the pseudospectral abscissa (Trefethen and Embree 2005), or via an H_∞ norm constraint (Zhou et al. 1996). If the uncertainty δ is structured, then we can try to optimize a performance criterion at every vertex in the polytopic description (which is a relaxation of the problem of stabilizing the whole polytope). An example is the problem of simultaneous stabilization, where a controller K must be found such that the maximum spectral abscissa of several matrices $A_i(K), i = 1, \ldots, N$ is negative (Blondel 1994). Finally, if the uncertainty δ is time varying, then performance and stability guarantees can still be achieved with the help of Lyapunov certificates or potentially conservative convex LMI conditions: see, e.g., Boyd et al. (1994) and Scherer et al. (1997).

A unified approach to addressing conflicting performance criteria and uncertainty consists of searching for locally optimal solutions of a nonsmooth optimization problem that is built to incorporate minimization objectives and constraints for multiple plants. This is called (linear robust) **multiobjective control**, and formally, it can be expressed as the following optimization problem

$$\min_K \max_{i=1,\ldots,N}\{g_i(K) : \beta_i = \infty\}$$
$$\text{s.t. } g_i(K) \leq \beta_i, i = 1, \ldots, N,$$

where each $g_i(K)$ is a function of the closed-loop matrix $A_i(K)$ (e.g., a spectral abscissa or an H_∞ norm) and the scalars β_i are given and such that if $\beta_i = \infty$ for some i, then g_i appears in the objective function and not in a constraint: see Gumussoy et al. (2009) for details. In the above problem, the objective function, a maximum of nonsmooth and nonconvex functions, is typically also nonsmooth and nonconvex. Moreover, without loss of generality,

we can easily impose a sparsity pattern on controller matrix K to account for structural constraints (e.g., a low-order decentralized controller).

Software Packages

Algorithms for **nonconvex nonsmooth optimization** have been developed and interfaced for linear robust multiobjective control in the public domain Matlab package HIFOO released in Burke et al. (2006a) and based on the theory described in Burke et al. (2006b). In 2011, The MathWorks released HINFSTRUCT, a commercial implementation of these techniques based on the theory described in Apkarian and Noll (2006).

Cross-References

▶ H-Infinity Control
▶ LMI Approach to Robust Control

Bibliography

Ackermann J (1993) Robust control – systems with uncertain physical parameters. Springer, Berlin
Apkarian P, Noll D (2006) Nonsmooth H-infinity synthesis. IEEE Trans Autom Control 51(1): 71–86
Blondel VD (1994) Simultaneous stabilization of linear systems. Springer, Heidelberg
Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia
Burke JV, Lewis AS, Overton ML (2001) Optimizing matrix stability. Proc AMS 129:1635–1642
Burke JV, Henrion D, Lewis AS, Overton ML (2006a) HIFOO – a Matlab package for fixed-order controller design and H-infinity optimization. In: Proceedings of the IFAC symposium robust control design, Toulouse
Burke JV, Henrion D, Lewis AS, Overton ML (2006b) Stabilization via nonsmooth, nonconvex optimization. IEEE Trans Autom Control 51(11):1760–1769
Chesi G (2010) LMI techniques for optimization over polynomials in control: a survey. IEEE Trans Autom Control 55(11):2500–2510
Gumussoy S, Henrion D, Millstone M, Overton ML (2009) Multiobjective robust control with HIFOO 2.0. In: Proceedings of the IFAC symposium on robust control design (ROCOND 2009), Haifa
Henrion D, Lasserre JB (2004) Solving nonconvex optimization problems – how GloptiPoly is applied to problems in robust and nonlinear control. IEEE Control Syst Mag 24(3):72–83
Scherer CW, Gahinet P, Chilali M (1997) Multi-objective output feedback control via LMI optimization. IEEE Trans Autom Control 42(7):896–911
Trefethen LN, Embree M (2005) Spectra and pseudospectra: the behavior of nonnormal matrices and operators. Princeton University Press, Princeton
Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Optimization-Based Control Design Techniques and Tools

Pierre Apkarian[1] and Dominikus Noll[2]
[1]DCSD, ONERA – The French Aerospace Lab, Toulouse, France
[2]Institut de Mathématiques, Université de Toulouse, Toulouse, France

Abstract

Structured output feedback controller synthesis is an exciting new concept in modern control design, which bridges between theory and practice insofar as it allows for the first time to apply sophisticated mathematical design paradigms like H_∞ or H_2 control within control architectures preferred by practitioners. The new approach to structured H_∞ control, developed during the past decade, is rooted in a change of paradigm in the synthesis algorithms. Structured design may no longer be based on solving algebraic Riccati equations or matrix inequalities. Instead, optimization-based design techniques are required. In this essay we indicate why structured controller synthesis is central in modern control engineering. We explain why non-smooth optimization techniques are needed to compute structured control laws, and we point to software tools which enable practitioners to use these new tools in high-technology applications.

O

Keywords

Controller tuning; H_∞ synthesis; Multi-objective design; Non-smooth optimization; Structured controllers; Robust control

Introduction

In the modern high-technology field of control, engineers usually face a large variety of concurring design specifications such as noise or gain attenuation in prescribed frequency bands, damping, decoupling, constraints on settling or rise time, and much else. In addition, as plant models are generally only approximations of the true system dynamics, control laws have to be robust with respect to uncertainty in physical parameters or with regard to un-modeled high-frequency phenomena. Not surprisingly, such a plethora of constraints present a major challenge for controller tuning, not only due to the ever-growing number of such constraints but also because of their very different provenience.

The dramatic increase in plant complexity is exacerbated by the desire that regulators should be as simple as possible, easy to understand and to tune by practitioners, convenient to hardware implement, and generally available at low cost. Such practical constraints explain the limited use of black-box controllers, and they are the driving force for the implementation of *structured* control architectures, as well as for the tendency to replace hand-tuning methods by rigorous algorithmic optimization tools.

Structured Controllers

Before addressing specific optimization techniques, we introduce some basic terminology for control design problems with structured controllers. A state-space description of the given P used for design is given as

$$P : \begin{cases} \dot{x}_P = A x_P + B_1 w + B_2 u \\ z = C_1 x_P + D_{11} w + D_{12} u \\ y = C_2 x_P + D_{21} w + D_{22} u \end{cases} \quad (1)$$

where A, B_1,... are real matrices of appropriate dimensions, $x_P \in \mathbb{R}^{n_P}$ is the state, $u \in \mathbb{R}^{n_u}$ the control, $y \in \mathbb{R}^{n_y}$ the measured output, $w \in \mathbb{R}^{n_w}$ the exogenous input, and $z \in \mathbb{R}^{n_z}$ the regulated output. Similarly, the sought output feedback controller K is described as

$$K : \begin{cases} \dot{x}_K = A_K x_K + B_K y \\ u = C_K x_K + D_K y \end{cases} \quad (2)$$

with $x_K \in \mathbb{R}^{n_K}$ and is called *structured* if the (real) matrices A_K, B_K, C_K, D_K depend smoothly on a design parameter $\mathbf{x} \in \mathbb{R}^n$, referred to as the vector of tunable parameters. Formally, we have differentiable mappings

$$A_K = A_K(\mathbf{x}), B_K = B_K(\mathbf{x}), C_K = C_K(\mathbf{x}),$$
$$D_K = D_K(\mathbf{x}),$$

and we abbreviate these by the notation $K(\mathbf{x})$ for short to emphasize that the controller is structured with $\mathbf{x}$ as tunable elements.

A structured controller synthesis problem is then an optimization problem of the form

$$\begin{aligned} & \text{minimize } \|T_{wz}(P, K(\mathbf{x}))\| \\ & \text{subject to } K(\mathbf{x}) \text{ closed-loop stabilizing} \quad (3) \\ & \qquad K(\mathbf{x}) \text{ structured, } \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

where $T_{wz}(P, K) = \mathcal{F}_\ell(P, K)$ is the lower feedback connection of (1) with (2) as in Fig. 1 (left), also called the linear fractional transformation (Varga and Looye 1999). The norm $\|\cdot\|$ stands for the H_∞ norm, the H_2 norm, or any other system norm, while the optimization variable $\mathbf{x} \in \mathbb{R}^n$ regroups the tunable parameters in the design.

Standard examples of structured controllers $K(\mathbf{x})$ include realizable PIDs and observer-based, reduced-order, or decentralized controllers, which in state space are expressed as

$$\left[\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & -1/\tau & -k_D/\tau \\ \hline k_I & 1/\tau & k_P + k_D/\tau \end{array} \right], \left[\begin{array}{c|c} A - B_2 K_c - K_f C_2 & K_f \\ \hline -K_c & 0 \end{array} \right],$$

$$\left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array} \right], \left[\begin{array}{c|c} \operatorname*{diag}_{i=1}^{q} A_{Ki} & \operatorname*{diag}_{i=1}^{q} B_{Ki} \\ \hline \operatorname*{diag}_{i=1}^{q} C_{Ki} & \operatorname*{diag}_{i=1}^{q} D_{Ki} \end{array} \right].$$
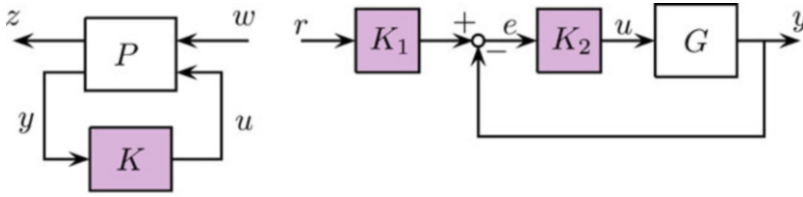
Optimization-Based Control Design Techniques and Tools, Fig. 1 Black-box full-order controller K on the *left*, structured 2-DOF control architecture with $K =$ block-diag(K_1, K_2) on the *right*

Optimization-Based Control Design Techniques and Tools, Fig. 2 Synthesis of $K =$ block-diag$(K_1, \ldots, K_N)$ against multiple requirements or models $P^{(1)}, \ldots, P^{(M)}$. Each $K_i(\mathbf{x})$ can be structured
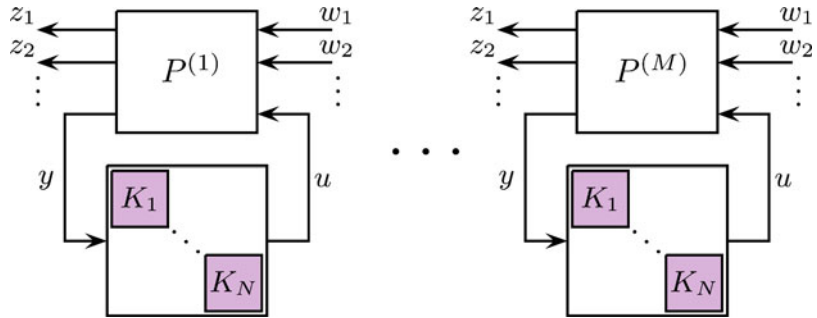
In the case of a PID, the tunable parameters are $\mathbf{x} = (\tau, k_P, k_I, k_D)$, for observer-based controllers $\mathbf{x}$ regroups the estimator and state-feedback gains (K_f, K_c), for reduced order controllers $n_K < n_P$ the tunable parameters $\mathbf{x}$ are the $n_K^2 + n_K n_y + n_K n_u + n_y n_u$ unknown entries in (A_K, B_K, C_K, D_K), and in the decentralized form $\mathbf{x}$ regroups the unknown entries in $A_{K1}, \ldots, D_{Kq}$. In contrast, full-order controllers have the maximum number $N = n_P^2 + n_P n_y + n_P n_u + n_y n_u$ of degrees of freedom and are referred to as unstructured or as *black-box* controllers.

More sophisticated controller structures $K(\mathbf{x})$ arise from architectures like, for instance, a 2-DOF control arrangement with feedback block K_2 and a set-point filter K_1 as in Fig. 1 (right). Suppose K_1 is the 1st-order filter $K_1(s) = a/(s + a)$ and K_2 the PI feedback $K_2(s) = k_P + k_I/s$. Then the transfer T_{ry} from r to y can be represented as the feedback connection of P and $K(\mathbf{x})$ with

$$P := \left[\begin{array}{c|ccc} A & 0 & 0 & B \\ \hline C & 0 & 0 & D \\ 0 & I & 0 & 0 \\ -C & 0 & I & -D \end{array} \right], K(\mathbf{x}) := \left[\begin{array}{cc} K_1(s) & 0 \\ 0 & K_2(s) \end{array} \right],$$

where $K(\mathbf{x})$ takes a typical block-diagonal structure featuring the tunable elements $\mathbf{x} = (a, k_P, k_I)$.

In much the same way, arbitrary multi-loop interconnections of fixed-model elements with tunable controller blocks $K_i(\mathbf{x})$ can be rearranged as in Fig. 2 so that $K(\mathbf{x})$ captures all tunable blocks in a decentralized structure general enough to cover most engineering applications.

The structure concept is equally useful to deal with the second central challenge in control design: *system uncertainty*. The latter may be handled with μ-synthesis techniques (Stein and Doyle 1991) if a parametric uncertain model is available. A less ambitious but often more practical alternative consists in optimizing the structured controller $K(\mathbf{x})$ against a finite set of plants $P^{(1)}, \ldots, P^{(M)}$ representing model variations due to uncertainty, aging, sensor and actuator breakdown, and un-modeled dynamics, in tandem with the robustness and performance specifications. This is again formally covered by Fig. 2 and leads to a multi-objective constrained optimization problem of the form

$$\text{minimize} \quad f(\mathbf{x}) = \max_{k \in \text{SOFT}, i \in I_k} \|T_{w_i z_i}^{(k)}(K(\mathbf{x}))\|$$

subject to $g(\mathbf{x}) = \max\limits_{k \in \text{HARD}, j \in J_k} \|T^{(k)}_{w_j z_j}(K(\mathbf{x}))\| \leq 1$

$$K(\mathbf{x}) \text{ structured and stabilizing}$$

$$\mathbf{x} \in \mathbb{R}^n \qquad (4)$$

where $T^{(k)}_{w_i z_i}$ denotes the ith closed-loop robustness or performance channel $w_i \rightarrow z_i$ for the kth plant model $P^{(k)}(s)$. The rationale of (4) is to minimize the worst-case cost of the soft constraints $\|T^{(k)}_{w_i z_i}\|$, $k \in \text{SOFT}$ while enforcing the hard constraints $\|T^{(k)}_{w_j z_j}\| \leq 1$, $k \in \text{HARD}$. Note that in the mathematical programming terminology, soft and hard constraints are classically referred to as objectives and constraints. The terms soft and hard point to the fact that hard constraints prevail over soft ones and that meeting hard constraints for solution candidates is mandatory.

Optimization Techniques Over the Years

During the late 1990s, the necessity to develop design techniques for structured regulators $K(\mathbf{x})$ was recognized (Fares et al. 2001), and the limitations of synthesis methods based on algebraic Riccati equations (AREs) or linear matrix inequalities (LMIs) became evident, as these techniques can only provide black-box controllers. The lack of appropriate synthesis techniques for structured $K(\mathbf{x})$ led to the unfortunate situation, where sophisticated approaches like the H_∞ paradigm developed by academia since the 1980s could not be brought to work for the design of those controller structures $K(\mathbf{x})$ preferred by practitioners. Design engineers had to continue to rely on heuristic and ad hoc tuning techniques, with only limited scope and reliability. As an example, post-processing to reduce a black-box controller to a practical size is prone to failure. It may at best be considered a fill-in for a rigorous design method which directly computes a reduced-order controller. Similarly, hand-tuning of the parameters $\mathbf{x}$ remains a puzzling task because of the loop interactions and fails as soon as complexity increases.

In the late 1990s and early 2000s, a change of methods was observed. Structured H_2- and H_∞-synthesis problems (3) were addressed by bilinear matrix inequality (BMI) optimization, which used local optimization techniques based on the augmented Lagrangian method (Fares et al. 2001; Noll et al. 2002; Kocvara and Stingl 2003), sequential semidefinite programming methods (Fares et al. 2002; Apkarian et al. 2003), and non-smooth methods for BMIs (Noll et al. 2009; Lemaréchal and Oustry 2000). However, these techniques were based on the bounded real lemma or similar matrix inequalities and were therefore of limited success due to the presence of Lyapunov variables, i.e., matrix-valued unknowns, whose dimension grows quadratically in $n_P + n_K$ and represents the bottleneck of that approach.

The epoch-making change occurs with the introduction of non-smooth optimization techniques (Noll and Apkarian 2005; Apkarian and Noll 2006b,c, 2007) to programs (3) and (4). Today non-smooth methods have superseded matrix inequality-based techniques and may be considered the state of the art as far as realistic applications are concerned. The transition took almost a decade.

Alternative control-related local optimization techniques and heuristics include the gradient sampling technique of Burke et al. (2005), derivative-free optimization discussed in Kolda et al. (2003) and Apkarian and Noll (2006a) and particle swarm optimization; see Oi et al. (2008) and references therein and also evolutionary computation techniques (Lieslehto 2001). The last three classes do not exploit derivative information and rely on function evaluations only. They are therefore applicable to a broad variety of problems including those where function values arise from complex numerical simulations. The combinatorial nature of these techniques, however, limits their use to small problems with a few tens of variable. More significantly, these methods often lack a solid convergence theory. In contrast, as we have demonstrated over recent years (Apkarian and Noll 2006b; Noll et al. 2008),

specialized non-smooth techniques are highly efficient in practice, are based on a sophisticated convergence theory, are capable of solving medium-size problems in a matter of seconds, and are still operational for large-size problems with several hundreds of states.
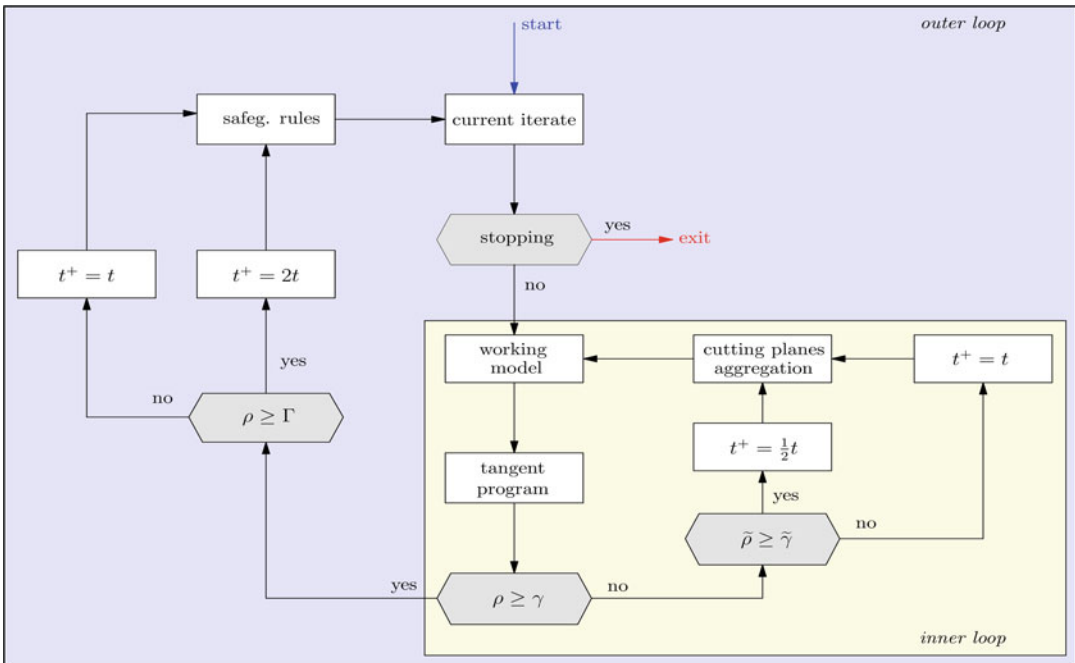
Non-smooth Optimization Techniques

The benefit of the non-smooth casts (3) and (4) lies in the possibility to avoid searching for Lyapunov variables, a major advantage as their number $(n_P + n_K)^2/2$ usually largely dominates n, the number of true decision parameters $\mathbf{x}$. Lyapunov variables do still occur implicitly in the function evaluation procedures, but this has no harmful effect for systems up to several hundred states. In abstract terms, a non-smooth optimization program has the form

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (5)$$

where $f, g : \mathbb{R}^n \rightarrow \mathbb{R}$ are locally Lipschitz functions and are easily identified from the cast in (4).

In the realm of convex optimization, non-smooth programs are conveniently addressed by so-called bundle methods, introduced in the late 1970s by Lemaréchal (1975). Bundle methods are used to solve difficult problems in integer programming or in stochastic optimization via Lagrangian relaxation. Extensions of the bundling technique to non-convex problems like (3) or (4) were first developed in Apkarian and Noll (2006b,c, 2007), Apkarian et al. (2008), Noll et al. (2009), and, in more abstract form, Noll et al. (2008).

Figure 3 shows a schematic view of a non-convex bundle method consisting of a descent-step generating inner loop (yellow block) comparable to a line search in smooth optimization, embedded into the outer loop

Optimization-Based Control Design Techniques and Tools, Fig. 3 Flowchart of proximity control bundle algorithm

(blue box), where serious iterates are processed, stopping criteria are applied, and the model tradition is assured. Serious steps or iterates refer to steps accepted in a line search, while null steps are unsuccessful steps visited during the search. By model tradition, we mean continuity of the model between (serious) iterates x^j and x^{j+1} by recycling some of the older planes used at counter j into the new working model at $j + 1$. This avoids starting the first inner loop $k = 1$ at $j + 1$ from scratch and therefore saves time.

At the core of the interaction between inner and outer loop is the management of the proximity control parameter τ, which governs the stepsize $\|\mathbf{x} - \mathbf{y}^k\|$ between trial steps $\mathbf{y}^k$ at the current serious iterate $\mathbf{x}$. Similar to the management of a trust region radius or of the stepsize in a line search, proximity control allows to force shorter trial steps if agreement of the local model with the true objective function is poor and allows larger steps if agreement is satisfactory.

Oracle-based bundle methods traditionally assure global convergence in the sense of subsequences under the sole hypothesis that for every trial point $\mathbf{x}$, the function value $f(\mathbf{x})$ and a Clarke subgradient $\phi \in \partial f(\mathbf{x})$ are provided. In automatic control applications, it is as a rule possible to provide more specific information, which may be exploited to speed up convergence.

Computing function value and gradients of the H_2 norm $f(\mathbf{x}) = \|T_{wz}(P, K(\mathbf{x}))\|_2$ requires essentially the solution of two Lyapunov equations of size $n_P + n_K$ (see Apkarian et al. 2007; Rautert and Sachs 1997). For the H_∞ norm, $f(\mathbf{x}) = \|T_{wz}(P, K(\mathbf{x}))\|_\infty$, function evaluation is based on the Hamiltonian algorithm of Benner et al. (2012) and Boyd et al. (1989). The Hamiltonian matrix is of size $n_P + n_K$ so that function evaluations may be costly for very large plant state dimension ($n_P > 500$), even though the number of outer loop iterations of the bundle algorithm is not affected by a large n_P and generally relates to n, the dimension of $\mathbf{x}$. The additional cost for subgradient computation for large n_P is relatively cheap as it relies on linear algebra (Apkarian and Noll 2006b).

Computational Tools

The novel non-smooth optimization methods became available to the engineering community since 2010 via the MATLAB Robust Control Toolbox (Robust Control Toolbox 4.2 2012; Gahinet and Apkarian 2011). Routines HINFSTRUCT, LOOPTUNE, and SYSTUNE are versatile enough to define and combine tunable blocks $K_i(\mathbf{x})$, to build and aggregate design requirements $T_{wz}^{(k)}$ of different nature, and to provide suitable validation tools. Their implementation was carried out in cooperation with P. Gahinet (MathWorks). These routines further exploit the structure of problem (4) to enhance efficiency (see Apkarian and Noll 2006b, 2007).

It should be mentioned that design problems with multiple hard constraints are inherently complex. It is well known that even simultaneous stabilization of more than two plants $P^{(j)}$ with a structured control law $K(\mathbf{x})$ is NP-complete so that exhaustive methods are expected to fail even for small to medium problems. The principled decision made in Apkarian and Noll (2006b) and reflected in the MATLAB routines is to rely on local optimization techniques instead. This leads to weaker convergence certificates but has the advantage to work successfully in practice. In the same vein, in (4) it is preferable to rely on a mixture of soft and hard requirements, for instance, by the use of exact penalty functions (Noll and Apkarian 2005). Key features implemented in the mentioned MATLAB routines are discussed in Apkarian (2013), Gahinet and Apkarian (2011), and Apkarian and Noll (2007).

Design Example

Design of a feedback regulator is an interactive process, in which tools like SYSTUNE, LOOPTUNE, or HINFSTRUCT support the designer in various ways. In this section we illustrate their enormous potential by solving a multi-model, fixed-structure reliable flight control design problem.
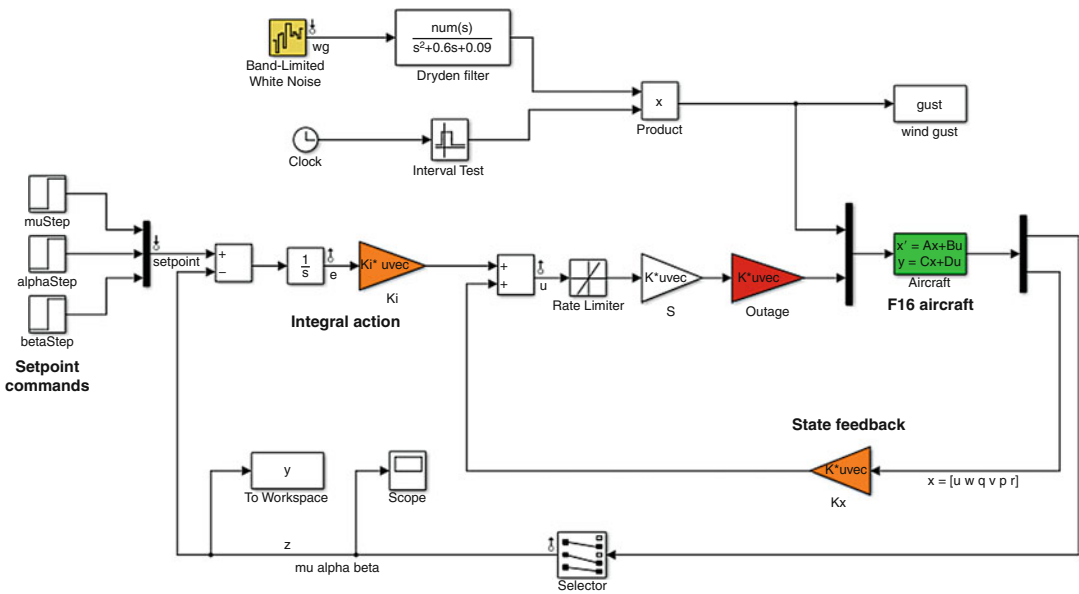
Optimization-Based Control Design Techniques and Tools, Fig. 4 Synthesis interconnection for reliable control

Optimization-Based Control Design Techniques and Tools, Table 1 Outage scenarios where 0 stands for failure

Outage cases	Diagonal of outage gain				
Nominal mode	1	1	1	1	1
Right elevator outage	0	1	1	1	1
Left elevator outage	1	0	1	1	1
Right aileron outage	1	1	0	1	1
Left aileron outage	1	1	1	0	1
Left elevator and right aileron outage	1	0	0	1	1
Right elevator and right aileron outage	0	1	0	1	1
Right elevator and left aileron outage	0	1	1	0	1
Left elevator and left aileron outage	1	0	1	0	1

In reliable flight control, one has to maintain stability and adequate performance not only in nominal operation but also in various scenarios where the aircraft undergoes outages in elevator and aileron actuators. In particular, wind gusts must be alleviated in all outage scenarios to maintain safety. Variants of this problem are addressed in Liao et al. (2002).

The open loop F16 aircraft in the scheme of Fig. 4 has six states, the body velocities u, v, w and pitch, roll, and yaw rates q, p, r. The state is available for control as is the flight-path bank angle rate μ (deg/s), the angle of attack α (deg), and the sideslip angle β (deg). Control inputs are the left and right elevator, left and right aileron,

and rudder deflections (deg). The elevators are grouped symmetrically to generate the angle of attack. The ailerons are grouped antisymmetrically to generate roll motion. This leads to three control actions as shown in Fig. 4. The controller consists of two blocks, a 3×6 state-feedback gain matrix K_x in the inner loop and a 3×3 integral gain matrix K_i in the outer loop, which leads to a total of $27 = \dim \mathbf{x}$ parameters to tune.

In addition to nominal operation, we consider eight outage scenarios shown in Table 1.

The different models associated with the outage scenarios are readily obtained by premultiplication of the aircraft control input by a diagonal matrix built from the rows in Table 1.
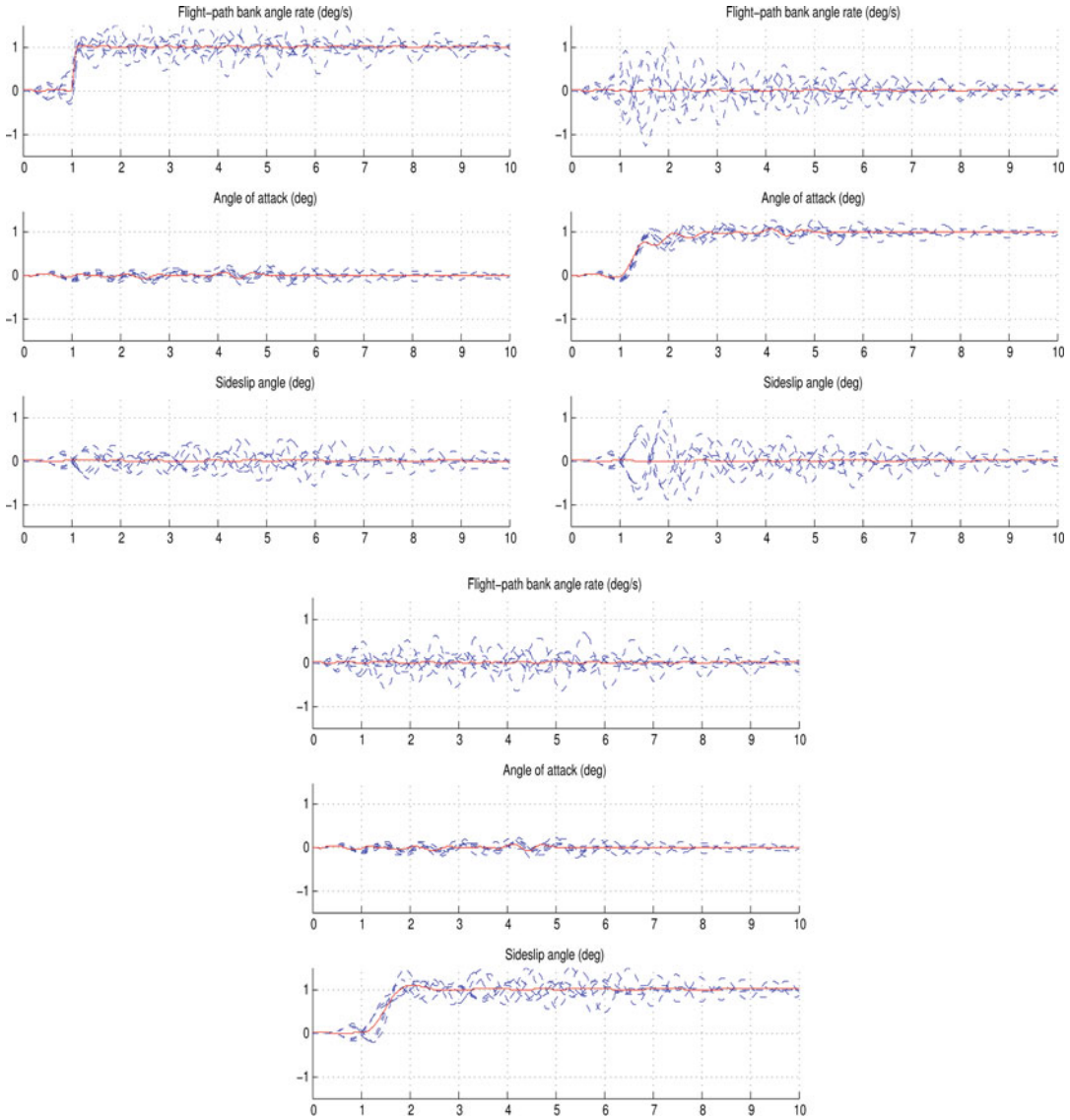
Optimization-Based Control Design Techniques and Tools, Fig. 5 Responses to step changes in μ, α, and β for nominal design

The design requirements are as follows:
- Good tracking performance in μ, α, and β with adequate decoupling of the three axes.
- Adequate rejection of wind gusts of 5 m/s.
- Maintain stability and acceptable performance in the face of actuator outage.

Tracking is addressed by an LQG cost (Maciejowski 1989), which penalizes integrated tracking error e and control effort u via

$$J = \lim_{T \to \infty} E \left(\frac{1}{T} \int_0^T \|W_e e\|^2 + \|W_u u\|^2 dt \right). \tag{6}$$

Diagonal weights W_e and W_u provide tuning knobs for trade-off between responsiveness, control effort, and balancing of the three channels. We use $W_e = \text{diag}(20, 30, 20)$, $W_u = I_3$ for normal operation and $W_e = \text{diag}(8, 12, 8)$, $W_u = I_3$ for outage conditions. Model-dependent weights
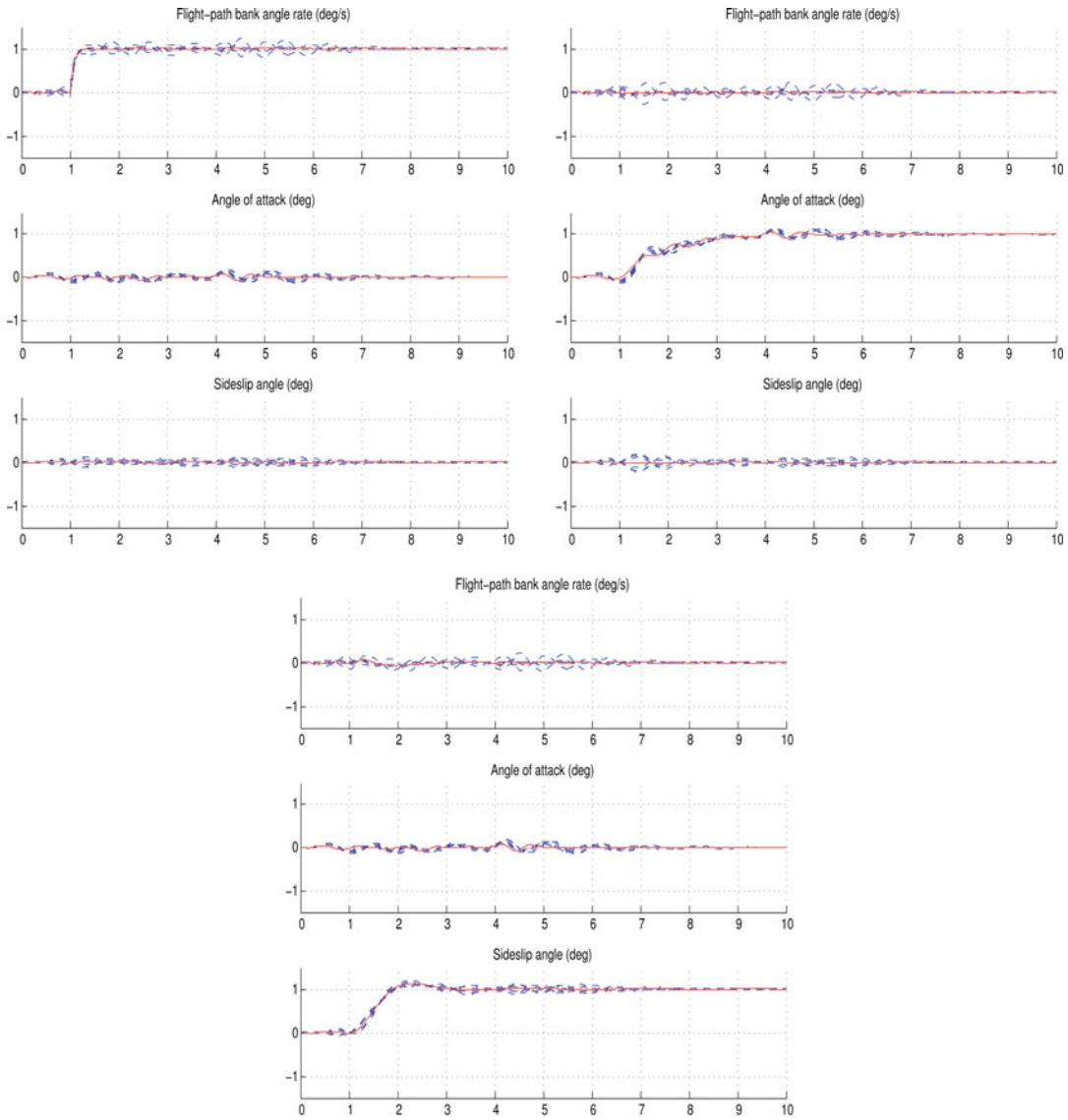
Optimization-Based Control Design Techniques and Tools, Fig. 6 Responses to step changes in μ, α, and β for fault-tolerant design

allow to express the fact that nominal operation prevails over failure cases. Weights for failure cases are used to achieve limited deterioration of performance or of gust alleviation under deflection surface breakdown.

The second requirement, wind gust alleviation, is treated as a hard constraint limiting the variance of the error signal e in response to white noise w_g driving the Dryden wind gust model.

In particular, the variance of e is limited to 0.01 for normal operation and to 0.03 for the outage scenarios.

With the notation of section "Non-smooth Optimization Techniques," the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ in (5) are $f(\mathbf{x}) := \max_{k=1,\dots,9} \|T_{rz}^{(k)}(\mathbf{x})\|_2$ and $g(\mathbf{x}) := \max_{k=1,\dots,9} \|T_{w_g e}^{(k)}(\mathbf{x})\|_2$, where r denotes the set-point inputs in μ, α, and β. The regulated output z is

$$z^T := \left[(W_e^{1/2} e)^T \ (W_u^{1/2} u)^T \right]^T,$$

with $\mathbf{x} = (\text{vec}(K_i), \text{vec}(K_x)) \in \mathbb{R}^{27}$. Soft constraints are the square roots of J in (6) with appropriate weightings W_e and W_u, hard constraints the RMS values of e, suitably weighted to reflect variance bounds of 0.01 and 0.03. These requirements are covered by the `Variance` and `WeightedVariance` options in Robust Control Toolbox 4.2 (2012).

With this setup, we tuned the controller gains K_i and K_x for the nominal scenario only (*nominal design*) and for all nine scenarios (*fault-tolerant design*). The responses to setpoint changes in μ, α, and β with a gust speed of 5 m/s are shown in Fig. 5 for the nominal design and in Fig. 6 for the fault-tolerant design. As expected, nominal responses are good but notably deteriorate when faced with outages. In contrast, the fault-tolerant controller maintains acceptable performance in outage situations. Optimal performance (square root of LQG cost J in (6)) for the fault-tolerant design is only slightly worse than for the nominal design (26 vs. 23). The non-smooth program (5) was solved with `SYSTUNE`, and the fault-tolerant design (9 models, 11 states, 27 parameters) took 30 s on Mac OS X with 2.66 GHz Intel Core i7 and 8 GB RAM. The reader is referred to Robust Control Toolbox 4.2 (2012) or higher versions, for further examples, and additional details.

Future Directions

From an application viewpoint, non-smooth optimization techniques for control system design and tuning will become one of the standard techniques in the engineer's toolkit. They are currently studied in major European aerospace industries.

Future directions may include:

- Extension of these techniques to gain scheduling in order to handle larger operating domains.
- Application of the available tools to integrated system/control when both system physical characteristics and controller elements are optimized to achieve higher performance. Application to fault detection and isolation may also reveal as an interesting vein.

Cross-References

- ▶ H-Infinity Control
- ▶ Optimization Based Robust Control
- ▶ Robust Synthesis and Robustness Analysis Techniques and Tools

Bibliography

Apkarian P (2013) Tuning controllers against multiple design requirements. In: American control conference (ACC), Washington, DC, pp 3888–3893

Apkarian P, Noll D (2006a) Controller design via nonsmooth multi-directional search. SIAM J Control Optim 44(6):1923–1949

Apkarian P, Noll D (2006b) Nonsmooth H_∞ synthesis. IEEE Trans Autom Control 51(1):71–86

Apkarian P, Noll D (2006c) Nonsmooth optimization for multidisk H_∞ synthesis. Eur J Control 12(3): 229–244

Apkarian P, Noll D (2007) Nonsmooth optimization for multiband frequency domain control design. Automatica 43(4):724–731

Apkarian P, Noll D, Thevenet JB, Tuan HD (2003) A spectral quadratic-SDP method with applications to fixed-order H_2 and H_∞ synthesis. Eur J Control 10(6):527–538

Apkarian P, Noll D, Rondepierre A (2007) Mixed H_2/H_∞ control via nonsmooth optimization. In: Proceedings of the 46th IEEE conference on decision and control, New Orleans, pp 4110–4115

Apkarian P, Noll D, Prot O (2008) A trust region spectral bundle method for nonconvex eigenvalue optimization. SIAM J Optim 19(1):281–306

Benner P, Sima V, Voigt M (2012) L_∞-norm computation for continuous-time descriptor systems using structured matrix pencils. IEEE Trans Autom Control 57(1):233–238

Boyd S, Balakrishnan V, Kabamba P (1989) A bisection method for computing the H_∞ norm of a transfer matrix and related problems. Math Control Signals Syst 2(3):207–219

Burke J, Lewis A, Overton M (2005) A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM J Optim 15:751–779

Fares B, Apkarian P, Noll D (2001) An augmented lagrangian method for a class of LMI-constrained problems in robust control theory. Int J Control 74(4): 348–360

Fares B, Noll D, Apkarian P (2002) Robust control via sequential semidefinite programming. SIAM J Control Optim 40(6):1791–1820

Gahinet P, Apkarian P (2011) Structured H_∞ synthesis in MATLAB. In: Proceedings of the IFAC world congress, Milan, pp 1435–1440

Kocvara M, Stingl M (2003) A code for convex nonlinear and semidefinite programming. Optim Methods Softw 18(3):317–333

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev 45(3):385–482

Lemaréchal C (1975) An extension of Davidon methods to nondifferentiable problems. In: Balinski ML, Wolfe P (eds) Nondifferentiable optimization. Mathematical programming study, vol 31. North-Holland, Amsterdam, pp 95–109

Lemaréchal C, Oustry F (2000) Nonsmooth algorithms to solve semidefinite programs. In: El Ghaoui L, Niculescu S-I (eds) SIAM advances in linear matrix inequality methods in control series. SIAM

Liao F, Wang JL, Yang GH (2002) Reliable robust flight tracking control: an LMI approach. IEEE Trans Control Syst Technol 10:76–89

Lieslehto J (2001) PID controller tuning using evolutionary programming. In: American control conference, Arlington, Virginia, vol 4, pp 2828–2833

Maciejowski JM (1989) Multivariable feedback design. Addison-Wesley, Wokingham

Noll D, Apkarian P (2005) Spectral bundle methods for nonconvex maximum eigenvalue functions: first-order methods. Math Program B 104(2):701–727

Noll D, Torki M, Apkarian P (2002) Partially augmented lagrangian method for matrix inequality constraints. Submitted Rapport Interne, MIP, UMR 5640, Maths. Dept. – Paul Sabatier University

Noll D, Prot O, Rondepierre A (2008) A proximity control algorithm to minimize nonsmooth and nonconvex functions. Pac J Optim 4(3):571–604

Noll D, Prot O, Apkarian P (2009) A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. J Convex Anal 16(3–4):641–666

Oi A, Nakazawa C, Matsui T, Fujiwara H, Matsumoto K, Nishida H, Ando J, Kawaura M (2008) Development of PSO-based PID tuning method. In: International conference on control, automation and systems, Seoul, Korea, pp 1917–1920

Rautert T, Sachs EW (1997) Computational design of optimal output feedback controllers. SIAM J Optim 7(3):837–852

Robust Control Toolbox 4.2 (2012) The MathWorks Inc., Natick

Stein G, Doyle J (1991) Beyond singular values and loopshapes. AIAA J Guid Control 14:5–16

Varga A, Looye G (1999) Symbolic and numerical software tools for LFT-based low order uncertainty modeling. In: Proceedings of the CACSD'99 symposium, Cohala, pp 1–6

Option Games: The Interface Between Optimal Stopping and Game Theory

Benoit Chevalier-Roignant[1] and Lenos Trigeorgis[2]

[1]Oliver Wyman, Munich, Germany

[2]University of Cyprus, Nicosia, Cyprus

Abstract

Managers can stake a claim by committing to capital investments today that can influence their rivals' behavior or take a "wait-and-see" or step-by-step approach to avoid possible adverse market consequences tomorrow. At the core of this corporate dilemma lies the classic trade-off between commitment and flexibility. This trade-off calls for a careful balancing of the merits of flexibility against those of commitment. This balancing is captured by option games.

Keywords

Game theory; Option games; Optimal stopping; Real options

Introduction

The global competitive environment has become increasingly more challenging as modern economies undergo unprecedented changes in the midst of the global economic turmoil. Real-world dilemmas corporate managers face today are driven by the interplay among strategic and market uncertainty. The tech industry has evolved most rapidly, putting companies unable to respond to market developments and technological breakthroughs at severe disadvantage. Corporate management's plans and how they implement their strategy will likely determine whether the firm will survive and be successful in the marketplace or become extinct.

Formulating the right strategy in the right competitive environment at the right time is a nontrivial task. Whether to invest in a new technology, a new product or enter a new market is a strategic decision of immense importance. Corporate management must assess strategic options with proper analytical tools that can help determine whether to commit to a particular strategic path, given scarce or costly resources, or whether to stay flexible. Oftentimes, firms need to position themselves flexibly to capitalize on future opportunities as they emerge, while limiting potential losses arising from adverse future circumstances. In many cases, corporate managers find themselves in need to revise their decision plans in view of actual market developments when facing an uncertain future; they can then decide to undertake only those projects with sufficiently high prospects in the future to justify commitment at that time. This needs to be balanced with the need to make irreversible strategic commitments to seize first-mover advantage presenting rivals with a *fait accompli* to which they have no choice but adapt.

Capital Budgeting Ignoring Strategic Interactions

Net Present Value

Prevailing management approaches simplify matters and often lead to investment decisions that are detrimental to the firm's long-term well-being. Suppose a firm's future cash flow at time t is given by a random variable X_t. Cash flows then evolve as a geometric Brownian motion

$$dX_t = gX_t dt + \sigma X_t dB_t \text{ and } X_0 \equiv x$$

with drift parameter g and volatility σ. The Brownian motion $(B_t; t \geq 0)$ captures exogenous market uncertainty. The standard criterion used in corporate finance is based on discounted cash flows (DCF) or net present value (NPV). This consists in assessing the current value of a project by discounting the expected future cash flows $E[X_t]$ at a constant discount rate, r. Management supposedly creates shareholder value by under-

taking projects with positive NPV, i.e., projects for which the present value of cash flows, $v(x) = \int_0^\infty e^{-rt} E[X_t] dt$, exceeds the necessary investment cost, I. In the present case, the firm will invest under the zero-NPV criterion if

$$\frac{x}{r-g} \geq I \qquad (1)$$

This traditional criterion views investment opportunities as now-or-never decisions under passive management. However, this precludes the possibility to adjust future decisions in case the market develops off the expected path. While market uncertainty is factored in through the discount rate, the flexibility management has is typically not properly accounted for.

Real Options Analysis

It has become standard practice in finance and strategy to interpret real investment opportunities as being analogous to financial options. This view is well accepted among academics and practitioners alike and is at the core of real options analysis (ROA). ROA is an extension of option-pricing theory to real investment situations (Myers 1977; Trigeorgis 1996). This approach effectively allows one to capture the dynamic nature of decision-making since it factors in management's flexibility to revise and adapt its decision in the face of market uncertainty. ROA allows managers with flexibility to adapt to actual market developments as uncertainty gets resolved. Managers may, for example, delay the start (or closure) of a project depending on its prospects. This approach leverages on optimal stopping theory (e.g., see Bensoussan and Lions 1982; Dixit and Pindyck 1994) and is considered to be more reflective of real decision-making than traditional methods. In the case the firm can delay the decision to invest, for example, the problem is one of optimal stopping:

$$V(x) = \max_T E[e^{-rT}(v(X_T) - I)]$$

by ROA, the discount rate r is the risk-free interest (Dixit and Pindyck 1994; Trigeorgis 1996). The time of managerial action, T, is

now a strategic decision variable random by nature as the decision maker faces an uncertain environment. This problem has an analytical solution characterized by a threshold policy, say a trigger $\bar{X}$, given by

$$\frac{\bar{X}}{r-g} = \frac{b}{b-1}I \qquad (2)$$

where b is the positive root of a quadratic function (e.g., see Dixit and Pindyck 1994) and $b/(b-1) > 1$. When decisions are costly or difficult to reverse, corporate managers would be more cautious and careful to make decisions. A firm should not always commit immediately – even if the NPV criterion (1) indicates so – but wait until the gross project value is sufficiently positive to cover the investment cost I by a factor larger than one, as expressed in (2). Investing prematurely may destroy shareholder value. Real options may justify sometimes undertaking projects with negative (static) net present value if it creates a platform for growth options or delaying projects with positive NPV.

Accounting for Strategic Interactions in Capital Budgeting

Strategic Uncertainty

As natural monopolies have lost their secular well-protected positions owing to market liberalization in the European Union and elsewhere across the globe, strategic interdependencies have become new key challenge for managers. At the same time sectors traditionally populated by multiple firms have undergone significant consolidation, often resulting in oligopolistic situations with a reduced number of players. The ongoing economic crisis has amplified these consolidation pressures. These two ongoing phenomena – liberalization and consolidation – have put high on the corporate agenda the assessment of strategic options under competition. Standard real options analysis often examines investment decisions as if the option holder has a proprietary right to exercise. This perspective may not be realistic in the new oligopolistic environment as several

firms may share the right to a related investment opportunity in the industry.

Game Theory

In oligopolistic industries, firms often have difficulty predicting how rivals will behave and make decisions based on beliefs about their likely behavior. A theory that helps characterize beliefs and form predictions about which strategies opponents will follow is helpful in analyzing such oligopolistic situations. Game theory has traditionally been used to frame strategic interactions arising in conflict situations involving parties with different objectives or interests. It attempts to model behavior in strategic situations or games in which one party's success in making choices depends on the choices of other players through influencing one another's welfare. Game theory adopts a different perspective on optimization, as the focus is on the formation of beliefs about how rivals' optimal strategies. Finance theory has been primarily concerned with "moves by nature," while game theory focuses on "optimization problems" involving multiple players. To solve a game, one needs to reduce a complex multiplayer problem into a simpler structure that captures the essence of the conflict situation. One can then derive useful predictions about how rivals are likely to react in a given situation. Game theory helped reshape microeconomics by providing analytical foundations for the study of market behavior and has been at the foundation of the Nobel prize winning research field of industrial organization.

Dynamic game theory (see, e.g., Basar and Olsder 1999) addresses problems in which several parties are in repeated interaction. Strategic management approaches based on dynamic economic theory can provide a richer foundation for understanding developments and competitive reactions within an industry. As firm competitiveness involves interactions among several players (rivals, suppliers or clients), game theoretic analysis brings important insights into strategic management in addressing such issues as first- and second-mover advantages, firm entry and exit decisions, strategic commitment, reputation, signaling, and other informational

O

effects. A key lesson is that, when firms react to one another, it may sometimes be appropriate for one firm to take an aggressive stance in expectation that rivals will back off. Dynamic industrial organization includes the analysis of "games of timing" such as preemption games or war of attrition, whereby firms decide on appropriate investment timing under rivalry.

Option Games

The earlier optimal stopping problem falls in the category of "games of timing" when a firm's entry decision influences another firm's market strategy. Option games are most suitable to help model situations where a firm that has a real option to (dis)invest faces rivalry. Here, the problem consists in finding a Nash equilibrium solution for the two-player equivalent of the above optimal stopping problem. This solution must also satisfy certain dynamic consistency criteria. For sequential investments, the follower is faced with a single-agent optimal investment timing problem; it will thus enter if the gross project value exceeds the investment cost by a sufficient factor. A firm entering the market early on, i.e., a leader, earns temporary monopoly rents as long as demand remains below the follower's entry threshold. Following the follower's entry, the firms act as a duopoly. As long as the leader's value exceeds the follower's, there is an incentive for one firm to invest, but not necessarily for both of them, leading to a "coordination problem." The competitive pressure will dissipate away the leader's first-mover advantage, leading to a market entry point that is not socially optimal and to rent dissipation. Unfortunately, the multiplayer problem does not involve a simple analytical solution, since at each point a duopolist firm might end up in any of four distinct situations (two-by-two matrix) depending on the rival's entry decision. Option games indicate in each situation which driving force (commitment vs. flexibility) prevails and whether to go ahead with the investment or wait and see. Main drivers of the prevailing market

equilibrium include the riskiness of the venture, σ, the magnitude of the first-mover advantage and the exclusive or shared ability to reap the benefits of the investment vis-à-vis rivals. When firms can grasp a large first-mover advantage from investing early but cannot differentiate themselves sufficiently from each other, they may be tempted to wage a preemptive war, investing prematurely at an early market stage that actually kills option value. If firms are more on an equal footing but do not see much benefit from investing early, they may prefer to wait and invest (jointly) at a later stage when the future market is sufficiently mature. If, however, one firm has a comparative cost advantage that dominates (e.g., a radical or drastic technological superiority) its rival industry, participants may prefer a consensual leader-follower investment arrangement involving less option value destruction.

Conclusions

Corporate management's strategic tool kit should provide clearer guidance on whether to pursue a wait-and-see stance in the face of uncertain market developments or jump on the first-mover bandwagon to build competitive advantage. We discussed two different modeling approaches that provide complementary perspectives and insights to help management deal with issues of flexibility versus commitment: real options and dynamic game theory. While each approach separately might turn a blind eye to flexibility or commitment, an integrative perspective through "options games" might provide the right balance and serve as a tool kit for adaptive competitive strategy. Both perspectives ultimately aim to derive better insights into industry dynamics under industry conditions characterized by both market and strategic uncertainty.

Option games pave the way for a consistent approach in addressing managerial decision-making, elevating the art of strategy to scientific analysis. Option games integrates in a common, consistent framework recent advances made in

these diverse set of disciplines. This emerging field that represents a promising strategic management tool that can help guide managerial decisions through the complexity of the modern competitive marketplace.

Cross-References

- ▶ Auctions
- ▶ Learning in Games

Recommended Reading

Smit and Trigeorgis (2004) discuss related trade-offs with discrete-time real option techniques. Grenadier (2000) and Huisman (2001) examine a number of continuous-time models. Chevalier-Roignant and Trigeorgis (2011) synthesize both types of "option games." An overview of the literature is provided in Chevalier-Roignant et al. (2011).

Bibliography

Basar T, Olsder GJ (1999) Dynamic noncooperative game theory, 2nd edn. SIAM, Philadelphia

Bensoussan A, Lions J-L (1982) Application of variational inequalities in stochastic control. North Holland, Amsterdam

Chevalier-Roignant B, Trigeorgis L (2011) Competitive strategy: options and games. MIT, Cambridge, MA

Chevalier-Roignant B, Flath CM, Huchzermeier A, Trigeorgis L (2011) Strategic investment under uncertainty: a synthesis. Eur J Oper Res 215(3):639–50

Dixit AK, Pindyck RS (1994) Investment under uncertainty. Princeton University Press, Princeton

Grenadier S (2000) Game choices: the intersection of real options and game theory. Risk Books, London

Huisman KJM (2001) Technology investment: a game theoretic real options approach. Springer, Boston

Myers SC (1977) Determinants of corporate borrowing. J Financ Econ 5(2):147–175

Smit HTJ, Trigeorgis L (2004) Strategic investment: real options and games. Princeton University Press, Princeton

Trigeorgis (1996) Real options. MIT, Cambridge, NA

Oscillator Synchronization

Bruce A. Francis
Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada

Abstract

The nonlinear Kuramoto equations for n coupled oscillators are derived and studied. The oscillators are defined to be synchronized when they oscillate at the same frequency and their phases are all equal. A control-theoretic viewpoint reveals that synchronized states of Kuramoto oscillators are locally asymptotically stable if every oscillator is coupled to all others. The problem of synchronization in Kuramoto oscillators is closely related to rendezvous, consensus, and flocking problems in distributed control. These problems, with their elegant solution by graph theory, are discussed briefly.

Keywords

Graph theory; Kuramoto model; Laplacian; Oscillator; Synchronization

Introduction

An oscillator is an electronic circuit or other kind of dynamical system that produces a periodic signal. If several oscillators are coupled together in some fashion and the periodic signals that they each produce are of the same frequency and are in phase, the oscillators are said to be synchronized. The book *Sync: The Emerging Science of Spontaneous Order*, by Strogatz, introduces a wide variety of phenomena where oscillators synchronize. Some examples from biology: networks of pacemaker cells in the heart, circadian pacemaker cells in the suprachiasmatic nucleus of the brain,
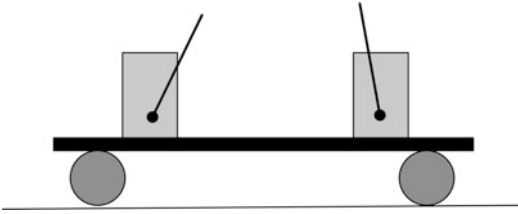
Oscillator Synchronization, Fig. 1 Two metronomes on a board that is on two pop cans. After the metronomes are let go at the same frequency but at different times, they soon become synchronized and tick in unison.

metabolic synchrony in yeast cell suspensions, groups of synchronously flashing fireflies, and crickets that chirp in unison. Engineering examples include clock synchronization in distributed communication networks and electric power networks with synchronous generators.

A very simple example of oscillator synchronization was discovered by Christiaan Huygens, the prominent Dutch scientist and mathematician who lived in the 1600s. One of his contributions was the invention of the pendulum clock, where a pendulum swings back and forth with a constant frequency. Huygens observed that two pendulum clocks in his house synchronized after some time. The explanation for this phenomenon is that the pendula were coupled mechanically through the wooden frame of the house. The same principle can be observed by a fun, simple experiment. As in Fig. 1, put two pop cans on a table, on their sides and parallel to each other. Place a board on top of them, and place two (or more) metronomes on the board. Set the metronomes to tick at the same frequency. Start them off ticking but not in unison. Within a few minutes they will be ticking in unison.

In this essay we derive what are known as the Kuramoto equations, a mathematical model of n oscillators, and then we study when they will synchronize.

The Kuramoto Model

In 1975 the Japanese researcher Yoshiki Kuramoto gave one of the first serious mathematical studies of coupled oscillators. To derive Kuramoto's equations, we begin with a simple hypothetical setup. Imagine n runners going around a circular track. Suppose they're all going at roughly the same speed, but each adjusts his/her speed based on the speeds of his/her nearest neighbors. If some runner passes another, that one tends to speed up to close the gap. The synchronization question is do the runners eventually end up running together in a tight pack?

Idealize the runners to be merely points, numbered $k = 1, \ldots, n$. They move on the unit circle in the complex plane. A point on the unit circle can be written as $e^{j\theta}$, where j denotes the unit imaginary number and θ denotes the angle measured counterclockwise from the positive real axis. The position of point k at time t is $z_k(t) = e^{j(\omega t + \theta_k(t))}$, where ω is the nominal rotational speed in rad/s, and $\theta_k(t)$ is the difference between the actual angle at time t and the nominal angle ωt. Notice that ω is a constant positive real number and it is the same for all n points. As in circuit theory, it simplifies the mathematics to refer all the positions to the sinusoid $e^{j\omega t}$, and therefore we define the **local position** of point k to be $p_k(t) = z_k(t)/e^{j\omega t}$, i.e., $p_k(t) = e^{j\theta_k(t)}$. Differentiate the local position with respect to time and let "dot" denote d/dt: $\dot{p}_k = e^{j\theta_k} j \dot{\theta}_k$. Define the local rotational velocity $v_k = \dot{\theta}_k$ and substitute into the preceding equation:

$$\dot{p}_k = v_k j p_k. \tag{1}$$

The local velocity v_k could be positive or negative. Notice that if we view p_k as a vector from the origin and view multiplication by j as rotation by $\pi/2$, then $j p_k$ can be viewed as tangent to the circle at the point p_k – see the picture on the left in Fig. 2.

Now we propose a feedback law for v_k in Eq. (1); see the picture on the right in Fig. 2. Take v_k proportional to the projection of p_i onto the tangent at p_k, that is, $v_k = \langle p_i, j p_k \rangle$. Here the inner product between two complex numbers v, w is $\langle v, w \rangle = \text{Re } \bar{v} w$. (You may check that this is equivalent to the usual dot product of vectors in $\mathbb{R}^2$.) Thus from (1) the model to get k to close the gap is $\dot{p}_k = \langle p_i, j p_k \rangle j p_k$.
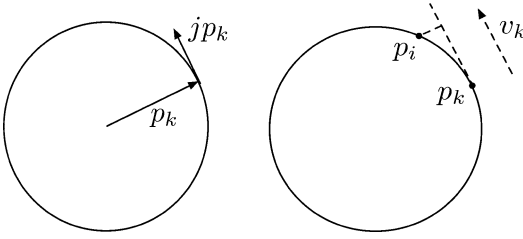
Oscillator Synchronization, Fig. 2 *Left*: The vectors p_k and jp_k. *Right*: The local velocity v_k

More generally, suppose that point k pays attention to not just point i but a fixed set of points called its **neighbors**. Let $\mathcal{N}_k$ denote the index set of neighbors of point k and for simplicity assume $\mathcal{N}_k$ does not depend on time. We consider the control law $v_k = \sum_{i \in \mathcal{N}_k} \langle p_i, jp_k \rangle$ and thereby arrive at the model of the evolution of the positions p_k:

$$\dot{p}_k = \sum_{i \in \mathcal{N}_k} \langle p_i, jp_k \rangle jp_k.$$

However, the Kuramoto model gives the evolution of the angles θ_k rather than the points p_k. To find the equation for θ_k, we observe that

$$\langle p_i, jp_k \rangle = \text{Re}\,(\bar{p}_i \, jp_k)$$
$$= \text{Re}\,(e^{-j\theta_i} j e^{j\theta_k})$$
$$= \sin(\theta_i - \theta_k).$$

In this way, the controlled points move according to

$$\dot{p}_k = \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k) jp_k.$$

Substitute in $p_k = e^{j\theta_k}$ and then cancel jp_k:

$$\dot{\theta}_k = \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k), \quad k = 1, \ldots, n. \quad (2)$$

This is the **Kuramoto model of coupled oscillators** in terms of the phases of the oscillators. There are n coupled nonlinear ordinary differential equations.

Equation (2) has the vector form $\dot{\theta} = g(\theta)$. There are some variations in the literature about

the state space associated with this equation. It is important to get the state space right because otherwise the concepts of stability and synchronization become shaky. The phase angles θ_k are real numbers with units of radians, so at first glance the state space is $\mathbb{R}^n$. But the angles are defined modulo 2π and so their values are restricted to lie in the interval $[0, 2\pi)$. In this way the state space becomes $[0, 2\pi)^n$. For example, if $n = 2$ the state space is the square $[0, 2\pi) \times [0, 2\pi)$ viewed as a subset of the plane $\mathbb{R}^2$. The mapping $\phi \mapsto e^{j\phi}$ is a one-to-one correspondence from the interval $[0, 2\pi)$ to the unit circle in the complex plane $\mathbb{C}$. This unit circle is usually denoted $\mathbb{S}^1$, the superscript signifying the circle's dimension as a manifold. By this correspondence the state space of (2) is the n-fold product $\mathbb{S}^1 \times \cdots \times \mathbb{S}^1$, and this is sometimes called the n-torus, denoted $\mathbb{T}^n$.

To recap, in what follows, the state space is $[0, 2\pi)^n$. This is an n-dimensional manifold rather than a vector space.

Synchronization

Control-theoretic methods, for example, that of Sepulchre et al. (2007), have been insightful. We address now the question of whether or not the oscillators in (2) synchronize, that is, the phases asymptotically converge to a common value. In the state space, $[0, 2\pi)^n$, the set of synchronized states is the set of vectors θ of the form $c\mathbf{1}$, where $c \in [0, 2\pi)$ and $\mathbf{1}$ is the vector of 1's. The simplest case is when every point is a neighbor of every other point, i.e., $\mathcal{N}_k$ contains every integer in the set $1, \ldots, n$ except k. Then (2) becomes

$$\dot{\theta}_k = \sum_{i=1}^{n} \sin(\theta_i - \theta_k), \quad k = 1, \ldots, n. \quad (3)$$

Let us show that if the initial phases $\theta_k(0)$ are all close enough together, then $\theta(t)$ converges asymptotically to a synchronized state. This will show that the synchronized states are locally asymptotically stable in a certain sense.

As stated before, Eq. (3) has the form $\dot{\theta} = g(\theta)$. The function $g(\theta)$ is the gradient of a

positive definite function. Indeed, let $re^{j\psi}$ denote the average of the points $e^{j\theta_1}, \ldots, e^{j\theta_n}$. Of course, r and ψ are functions of θ, and so we have

$$r(\theta)e^{j\psi(\theta)} = \frac{1}{n}\left(e^{j\theta_1} + \cdots + e^{j\theta_n}\right)$$

and therefore

$$r(\theta) = \frac{1}{n}\left|e^{j\theta_1} + \cdots + e^{j\theta_n}\right|.$$

The average of n points on the unit circle lives inside the unit disc, and therefore $r(\theta)$ is a real number between 0 and 1. It equals 1 if and only if the n points are equal, that is, the n phases are equal, and this is the state where the phases are synchronized.

Define the function

$$\begin{aligned}
V(\theta) &= \frac{n^2}{2}r(\theta)^2 \\
&= \frac{1}{2}\left|e^{j\theta_1} + \cdots + e^{j\theta_n}\right|^2 \\
&= \frac{1}{2}\left(e^{j\theta_1} + \cdots + e^{j\theta_n}\right)\left(e^{-j\theta_1} + \cdots + e^{-j\theta_n}\right).
\end{aligned}$$

Thus

$$\frac{\partial V(\theta)}{\partial \theta_k} = \sin(\theta_1 - \theta_k) + \cdots + \sin(\theta_n - \theta_k)$$

and therefore (3) can be written as $\dot{\theta} = \partial V(\theta)/\partial \theta$. This is a gradient equation. If $\theta(0)$ is chosen so that all the phases are close enough together, then $r(\theta(0))$ will be close to 1, and therefore θ will move in a direction to increase $V(\theta)$, that is, increase $r(\theta)$, until in the limit $r(\theta) = 1$ and the phases are synchronized.

There are results, e.g., Sepulchre et al. (2008), when the coupling is not all-to-all. Also, the term "synchronization" is used more generally than just for oscillators Wieland et al. (2011).

Rendezvous, Consensus, Flocking, and Infinitely Many Oscillators

Synchronization of coupled oscillators is closely related to other problems known as rendezvous,

consensus, or flocking problems. Phase synchronization is replaced by the requirement of mobile robots gathering at some location, by the requirement of temperature sensors in a sensor network converging to the same temperature estimate, or by the requirement that mobile robots should head in the same direction. The simplest form of these problems has the equations

$$\dot{\theta}_k = \sum_{i \in \mathcal{N}_k}(\theta_i - \theta_k), \quad k = 1, \ldots, n. \quad (4)$$

Notice that this can be obtained from the Kuramoto model (2) merely by replacing $\sin(\theta_i - \theta_k)$ by $\theta_i - \theta_k$ in (2), that is, by linearizing the latter at a synchronized state. We shall continue to call θ_k a phase of an oscillator. When do the phases evolving according to (4) synchronize? The answer to the question involves a lovely collaboration between graph theory and dynamics.

Introduce a directed graph that is in one-to-one correspondence with the neighbor structure. The graph is made up of n nodes, one for each oscillator. From each node there is an arrow to every neighbor of that node; that is, from node k is an arrow to every node in $\mathcal{N}_k$. Denote the adjacency matrix and the degree matrix of the graph by, respectively, A and D. That is, $a_{ij} = 1$ if j is a neighbor of i and d_{ii} equals the sum of the elements on row i of A. The **Laplacian** of the graph is defined to be $L = D - A$. Then (4) is equivalent to simply

$$\dot{\theta} = -L\theta, \quad (5)$$

where θ is still the vector with elements $\theta_1, \ldots, \theta_n$. Whether or not synchronization occurs depends on the connectivity of the graph. We stop here and refer the reader to the articles ▶ Averaging Algorithms and Consensus and ▶ Flocking in Networked Systems

Suppose there are an infinite but countable number of oscillators in the model (5). When will they synchronize? To answer this, we have to be more specific.

Let us allow an infinite number of oscillators numbered by the integers, positive, zero, and negative. Denote the phases by θ_k and let θ

denote the phase vector, whose kth component is θ_k. Assume each oscillator has only finitely many neighbors, let $\mathcal{N}_k$ denote the set of neighbors of oscillator k, and let L be the Laplacian of the associated graph. Finally, let $\theta(t)$ evolve according to the Eq. (5). This equation isn't automatically well posed in the sense that there may not be a solution defined for all $t > 0$. We have to impose a framework so that solutions do indeed exist. One natural space in which to place $\theta(0)$ is ℓ^2, the Hilbert space of square-summable sequences. If L is a bounded operator on ℓ^2, then so is e^{-Lt} for every $t > 0$, and hence the phase vector exists and belongs to ℓ^2 for every $t > 0$. Another natural space in which to place $\theta(0)$ is ℓ^∞, the Banach space of bounded sequences. Again, a phase vector exists for all $t > 0$ if L is a bounded operator on ℓ^∞.

The following example is from Feintuch and Francis (2012). Take the neighbor sets to be $\mathcal{N}_k = \{k - 1\}$. The graph is a chain: There is an arrow from node k to node $k - 1$, for every k, and the Laplacian is the infinite matrix with 1 on the diagonal, -1 on the first subdiagonal, and zero elsewhere. This Laplacian is a bounded operator on both ℓ^2 and ℓ^∞. Now the vector $c\mathbf{1}$, where $\mathbf{1}$ is the vector of all 1's, belongs to ℓ^∞ for every real number c, but it belongs to ℓ^2 only for $c = 0$. So the phases can potentially synchronize at any value in ℓ^∞, but only at 0 in ℓ^2. For the example under discussion, if the initial phase vector is in ℓ^2, then the phases synchronize at 0. By contrast, there exist initial phase vectors in ℓ^∞ such that synchronization does not occur. Even worse, $\lim_{t \to \infty} \theta(t)$ does not exist. The conclusion is that whether or not the oscillators will synchronize is a difficult question in general.

Summary and Future Directions

The Kuramoto model is a widely used paradigm for coupled oscillators. The model has the form $\dot{\theta} = f(E\theta)$, where θ is the vector of phases, the matrix E maps θ into the vector of possible differences $\theta_i - \theta_k$, and f is a function. The Kuramoto model considered in this essay is not the most general. A more general model allows different frequencies ω_k instead of just one, and also a coupling gain K, leading to the model

$$\dot{\theta}_k = \omega_k + \frac{K}{n} \sum_{i \in \mathcal{N}_k} \sin(\theta_i - \theta_k), \quad k = 1, \ldots, n.$$
(6)

An important problem associated with the Kuramoto model is to determine which synchronized states are stable. The linearized equation is interesting in its own right and relates to problems of rendezvous, consensus, and flocking.

Reference Dörfler and Bullo (2014) offers some questions for future study. In particular, it would be interesting to extend the Kuramoto model beyond the first-order oscillators of (2). Also, the case of general neighbor sets has much room for exploration.

Asymptotic stability is a robust property. For example, if the origin is asymptotically stable for the system $\dot{x} = Ax$, it remains so if A is perturbed by a sufficiently small amount. This is because the spectrum of a matrix is a continuous function of the matrix. The sketch in Fig. 1 vividly depicts the concept of synchronized oscillators. A topic for future study is that of robustness. Mathematically, if the two metronomes are identical, they will synchronize perfectly – this can be proved. Of course, physically two metronomes cannot be identical, and yet they will synchronize if they are close enough physically. A mathematical study of this phenomenon might be interesting.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Flocking in Networked Systems
▶ Graphs for Modeling Networked Interactions
▶ Networked Systems
▶ Vehicular Chains

Recommended Reading

The literature on the Kuramoto model is huge – there are now many hundreds of journal

papers continuing the study of oscillators using Kuramoto's model. There is space here only to highlight a few sources.

You can find a mathematical study of coupled metronomes in Pantaleone (2002). Also, Pantaleone's webpage Pantaleone describes some experimental observations. Kuramoto's original paper is Kuramoto (1975). Dörfler and Bullo have recently written a comprehensive survey (Dörfler and Bullo (2014)). Strogatz has written extensively on oscillator synchronization. His book *Sync* is fascinating and is highly recommended (Strogatz 2004). See also Strogatz (2000) and Strogatz and Stewart (1993). The papers Scardovi et al. (2007) and Dörfler and Bullo (2011) are recommended for more recent results, the latter treating the general model (6).

Getting phases in oscillators to synchronize is a special case of getting the states or outputs of coupled systems asymptotically to converge to a common value. There is a very large number of references on these subjects, a seminal one being Jadbabaie et al. (2003); others are Lin et al. (2007) and Moreau (2005). Regarding infinitely many oscillators, the physics literature treats only a continuum of oscillators, whereas countably many oscillators are the subject of Feintuch and Francis (2012).

Bibliography

Dörfler F, Bullo F (2011) On the critical coupling for Kuramoto oscillators. SIAM J Appl Dyn Syst 10(3):1070–1099

Dörfler F, Bullo F (2014) Synchronization in complex networks of phase oscillators: a survey. Automatica, 50(6), June 2014. To appear.

Feintuch A, Francis B (2012) Infinite chains of kinematic points. Automatica 48:901–908

Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans Automatic Control 48(6):988–1001

Kuramoto Y (1975) Self-entrainment of a population of coupled nonlinear oscillators. In: Araki H (ed) Volume 39 of International symposium on mathematical problems in theoretical physics, Kyoto. Lecture Notes in Physics. Springer, p 420

Lin Z, Francis BA, Maggiore M (2007) State agreement for coupled nonlinear systems with time-varying interaction. SIAM J Control Optim 46:288–307

Moreau L (2005) Stability of multi-agent systems with time-dependent communication links. IEEE Trans Automatic Control 50:169–182

Pantaleone J. Webpage. http://salt.uaa.alaska.edu/jim/

Pantaleone J (2002) Synchronization of metronomes. Am J Phys 70(10):992–1000

Scardovi L, Sarlette A, Sepulchre R (2007) Synchronization and balancing on the N-torus. Syst Control Lett 56:335–341

Sepulchre R, Paley DA, Leonard NE (2007) Stabilization of planar collective motion: all-to-all communication. IEEE Trans Automatic Control 52(5):811–824

Sepulchre R, Paley DA, Leonard NE (2008) Stabilization of planar collective motion with limited communication. IEEE Trans Automatic Control 53(3):706–719

Strogatz SH (2000) From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. Physica D 143:1–20

Strogatz SH (2004) Sync: the emerging science of spontaneous order. Hyperion Books, New York

Strogatz SH, Stewart I (1993) Coupled oscillators and biological synchronization. Sci Am 269:102–109

Wieland P, Sepulchre R, Allgower F (2011) An internal model principle is necessary and sufficient for linear output synchronization. Automatica 47:1068–1074

Output Regulation Problems in Hybrid Systems

Sergio Galeani

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma "Tor Vergata", Roma, Italy

Abstract

This entry discusses some of the salient features of the output regulation problem for hybrid systems, especially in connection with the steady-state characterization. In order to better highlight such peculiarities, the discussion is mostly focused on the simplest class of linear time-invariant systems exhibiting such behaviors. In comparison with the usual regulation theory, the role played by the zero dynamics and by the presence of more inputs than outputs is particularly striking.

Keywords

Disturbance rejection; Hybrid systems; Internal model principle; Output regulation; Tracking; Zero dynamics

Introduction

Output regulation is one of the most classical problems in control theory, and its celebrated solution in the linear time-invariant case (Davison 1976; Francis and Wonham 1976) is characterized by remarkable elegance and ideas (like the internal model principle). While the extension to nonlinear systems is still an active field of investigation, the study of output regulation for hybrid systems is also being actively pursued, and several surprising results have already appeared for the linear case, suggesting that a richer structure arises in hybrid output regulation problems due to the interplay between flow and jump dynamics.

The problem can be stated as follows. A known *exosystem* $\mathcal{E}$ with initial state belonging to a suitably defined set $\mathcal{W}_0$ produces a signal w possibly affecting both the *plant* $\mathcal{P}$ and the *compensator* $\mathcal{C}$; the compensator has to guarantee that for any initial state of $\mathcal{E}$ in a set $\mathcal{W}_0$:

- All closed-loop responses are bounded.
- The output e of $\mathcal{P}$ asymptotically converges to zero.

In order to avoid trivialities, the *exosystem* $\mathcal{E}$ is assumed to be such that its state evolution from nonzero initial states in $\mathcal{W}_0$ is bounded and not asymptotically converging to zero, both in forward and in backward time.

Two typical embodiments of the output regulation problem are the *disturbance rejection* and the *reference tracking* problems. In *disturbance rejection*, w acts as a disturbance on $\mathcal{P}$ and cannot be measured by $\mathcal{C}$, and the output e from which the effect of w has to be canceled is the actual plant output. In *reference tracking*, w contains the references to be tracked by an output y_r of $\mathcal{P}$, so that w can be assumed to be known by $\mathcal{C}$; by defining the regulated output e as $e = y_r - r$, the reference tracking problem is cast as an output regulation problem.

The solution of an output regulation problem entails the solution of two subproblems: the definition of a set of *zero output steady-state solutions* and the *asymptotic stabilization* of such solutions (or at least making them *attractive*; in many cases of interest, the achievement of this last objective actually yields asymptotic stabilization). As a matter of fact, the stabilization subproblem is already widely studied and described per se; for this reason, after some short remarks in section "Stabilization Obstructions in Hybrid Regulation", the remainder of this presentation will focus only on steady-state-related issues, for the simplest class of systems which exhibit the most peculiar and interesting phenomena of hybrid steady-state behavior (see in particular section "Key Features in Hybrid vs Classical Output Regulation"). For concreteness, only hybrid systems $\mathcal{E}, \mathcal{P}$ characterized by linear time-invariant (flow and jump) dynamics will be considered; following Goebel et al. (2012, Chap. 2), a two-dimensional parameterization of hybrid time $(t, k) \in \mathbb{R} \times \mathbb{N}$ will be used, with t measuring the flow of (usual) time and k counting the number of jumps experienced by the solution (see Fig. 1 for a specific example). So, the exosystem $\mathcal{E}$ will be described at time (t, k) by

$$\dot{w} = Sw, \qquad (w, t, k) \in \mathcal{C}_{\mathcal{E}}, \qquad (1a)$$

$$w^+ = Jw, \qquad (w, t, k) \in \mathcal{D}_{\mathcal{E}}, \qquad (1b)$$
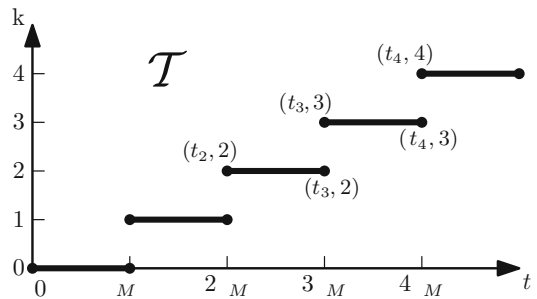
Output Regulation Problems in Hybrid Systems, Fig. 1 Hybrid time domain $\mathcal{T}$ for a "sampled data" hybrid system. *Dots* indicate $(t, k) \in \mathcal{T}$ when jumps occur (see section "Hybrid Steady-State Generation" for the t_k notation)

and the plant $\mathcal{P}$ will be described at time (t, k) by

$$\dot{x} = Ax + Bu + Pw, \quad (x, u, t, k) \in \mathcal{C}_{\mathcal{P}}, \tag{2a}$$

$$x^+ = Ex + Rw, \qquad (x, u, t, k) \in \mathcal{D}_{\mathcal{P}}, \tag{2b}$$

$$e = Cx + Qw, \tag{2c}$$

with $x(t, k) \in \mathbb{R}^n$, $u(t, k) \in \mathbb{R}^m$, $e(t, k) \in \mathbb{R}^p$, $w(t, k) \in \mathbb{R}^q$, and suitably defined flow sets $\mathcal{C}_{\mathcal{E}}$, $\mathcal{C}_{\mathcal{P}}$ and jump sets $\mathcal{D}_{\mathcal{E}}$, $\mathcal{D}_{\mathcal{P}}$.

Stabilization Obstructions in Hybrid Regulation

The achievement of asymptotic stabilization of the desired (zero output) steady-state responses for the considered class of linear hybrid systems crucially depends on whether the plant $\mathcal{P}$ and the exosystem $\mathcal{E}$ have synchronous jump times or not.

Asynchronous Jumps

Typically, jumps in $\mathcal{P}$ and $\mathcal{E}$ will be asynchronous, and this will cause the undesirable phenomenon that genuinely close trajectories will look "distant" around each jump when the distance is measured according to the usual Euclidean norm. The simplest illustration of such phenomenon consists in considering two trajectories of the same system starting from ε-close initial conditions. Consider the system

$$\dot{v} = 1, \quad v \in [0, 1], \qquad v^+ = 0, \quad v \notin (0, 1),$$

with the initial states $v_0 = 0$ and $v_1 = \varepsilon$, $0 < \varepsilon < 1$. The two ensuing solutions at time (t, k) are immediately computed as

$$v(t, k; v_0) = t - k, \qquad t \in [k, k + 1],$$

$$v(t, k; v_1) = \begin{cases} t - k + \varepsilon, & t \in [k, k + 1 - \varepsilon], \\ t - k + \varepsilon - 1, & t \in [k + 1 - \varepsilon, k + 1], \end{cases}$$

Hence, the (Euclidean) distance between the two solutions at time (t, k) is given by

$$d(t, k) = \begin{cases} \varepsilon, & t \in [k, k + 1 - \varepsilon], \\ (1 - \varepsilon), & t \in [k + 1 - \varepsilon, k + 1]; \end{cases}$$

in other words, choosing $\varepsilon > 0$ as small as desired, arbitrarily close initial conditions generate trajectories which are apart by a finite amount (as close as desired to 1) during the arbitrarily small time intervals where $t \in [k + 1 - \varepsilon, k + 1]$. Since stability deals with trajectories remaining close forever and attractivity deals with trajectories getting closer and closer, examples such as the one above pose serious issues when defining (let alone establish) stability and attractivity in the hybrid case. Similar problems arise not only in output regulation problems but also in other areas like state tracking, observers, and general interconnections of hybrid systems.

However, intuition suggests (and mathematics confirms, by using a suitable notion of "distance") that such trajectories are close indeed. Several approaches have been proposed in order to overcome such difficulty. Considering as an example a bouncing ball tracking another bouncing ball, the problematic time intervals are those between the bounce of the first ball hitting the ground and the bounce of the other ball; in such a case, the modified distances are defined by either

- Allowing to exclude sufficiently short "problematic" intervals (possibly requiring that their length asymptotically tends to zero); see, e.g., Galeani et al. (2008, 2012)
- Considering alternative "mirrored" trajectories computed as if the last jump did not happen; see, e.g., Forni et al. (2013a,b)
- Using a "stretched" distance function δ such that when point a is in the jump set and its image via the jump map is $g(a)$, then $\delta(a, b) = \delta(g(a), b)$; see, e.g., Biemond et al. (2013).

While the first approach has been proposed first, the other two (which are strongly related) have the advantage of providing (under mild additional hypotheses) global control Lyapunov functions.

Finally, it is worth noting that the most adequate tools to address similar issues for general hybrid systems are the "graphical distance" among hybrid arcs and related concepts (see Goebel et al. 2012, Chap. 5).

Synchronous Jumps

When synchronous jumps are considered, the above issue disappears, and asymptotic stabilization becomes a much simpler matter. Although synchronous jumps look more like an exception than a rule in hybrid systems, they are very reasonable for specific classes of problems.

In order to have synchronous jumps, some authors have considered the use of "jump inputs" which *impose a jump at a certain time*, which can be physically reasonable in some systems, e.g., two tanks separated by a movable wall, assuming that when the wall is removed the fluid reaches the equilibrium configuration almost instantaneously.

Another relevant class consists of "sampled data" systems, whose jumps are essentially due to digital components which operate at a fixed sampling rate, which will be considered in the rest of this entry. In such a case, letting τ_M be the sampling period, the time domain of the hybrid system is fixed as (see Fig. 1)

$$\mathcal{T} := \{(t,k) : t \in [k\tau_M, (k+1)\tau_M], k \in \mathbb{Z}\}, \tag{3}$$

all jumps happen exactly for (t,k) with $t = (k+1)\tau_M$, and then (1) can be simplified as

$$\dot{w} = Sw, \tag{4a}$$
$$w^+ = Jw, \tag{4b}$$

and (2) can be simplified as

$$\dot{x} = Ax + Bu + Pw, \tag{5a}$$
$$x^+ = Ex + Rw, \tag{5b}$$
$$e = Cx + Qw, \tag{5c}$$

since flow and jump times are clear from the context.

For the latter class of systems, by using linear time-invariant hybrid control laws and observers

(and an easily provable separation principle), it is easily shown that:

- Under a hybrid stabilizability hypothesis, state feedback stabilization of (5) is easily achieved.
- Output feedback stabilization of (5) from e is also trivial under an additional hybrid detectability hypothesis.
- Under hybrid detectability of the cascade of (4) and (5), w can be asymptotically estimated from e.

Due to the above facts, it can be assumed without loss of generality that (5) is asymptotically stable (equivalently, that all eigenvalues of $Ee^{A\tau_M}$ have modulus strictly less than one). Asymptotic stability then yields *incremental stability*, since letting $\hat{x}$ and $\check{x}$ denote two motions under the same inputs u, w and only differing in their initial states, it is immediate to see that their difference $\tilde{x} := \hat{x} - \check{x}$ evolves as

$$\dot{\tilde{x}} = \dot{\hat{x}} - \dot{\check{x}} = A\hat{x} + Bu + Pw$$
$$- (A\check{x} + Bu + Pw),$$
$$\tilde{x}^+ = \hat{x}^+ - \check{x}^+ = E\hat{x} + Rw - (E\check{x} + Rw),$$

that is, $\dot{\tilde{x}} = A\tilde{x}$, $\tilde{x}^+ = E\tilde{x}$, and so it is just a free motion of the plant, asymptotically converging to zero. Incremental stability implies that *regulation is achieved as soon as it is shown that for any exogenous input w it is possible to find an input u and an initial state of* (5) *such that e is identically zero*, since then any other motion arising from a different initial state will asymptotically converge to the motion with identically zero e. Moreover, it is easy to see that asymptotic stability of the origin actually implies uniform, global, and exponential stability of any trajectory for such systems.

Hybrid Steady-State Generation

From this point on, the rest of the presentation will be focused only on the case where the problem data are of the form (3) to (5), since this allows to provide an uncluttered view on some

peculiar features of hybrid steady-state motions, without the burden of having to take care of delicate stability issues arising in more general contexts.

Based on the preceding discussion, there is no loss of generality at this point in assuming that:

- *Plant* (5) *is asymptotically stable*, which is equivalent to all eigenvalues of $Ee^{A\tau_M}$ having a magnitude strictly less than one.
- *Exosystem* (4) *is Poisson stable*, which is equivalent to all eigenvalues of $Je^{S\tau_M}$ having a magnitude equal to one.

It is also customary to distinguish between *full information* and *error feedback* regulation, where in the first case controller $\mathcal{C}$ has access to the complete state (w, x) of the cascade of $\mathcal{E}$ and $\mathcal{P}$, whereas in the second case $\mathcal{C}$ can only measure the output e of $\mathcal{P}$.

Having assumed asymptotic stability of plant $\mathcal{P}$, the only role of compensator $\mathcal{C}$ consists in generating the correct steady-state input, since then, by incremental stability of $\mathcal{P}$, asymptotic regulation is ensured from any initial state. Recalling the expression of $\mathcal{T}$ in (3), for the following developments it is useful to define the jump times t_k and the elapsed time of flow since last jump σ as

$$t_k := k\tau_M, \qquad \sigma(t, k) := t - k\tau_M;$$

the arguments of $\sigma(t, k)$ will usually be omitted since clear from the context. Note that σ satisfies $\dot{\sigma} = 1$, $\sigma^+ = 0$, and it is often explicitly introduced as an additional *timer* variable.

The Full Information Case

Consider the candidate steady-state motion and input:

$$\begin{bmatrix} x_{ss}(t, k) \\ u_{ss}(t, k) \end{bmatrix} = \begin{bmatrix} \Pi(\sigma) \\ \Gamma(\sigma) \end{bmatrix} w(t, k). \qquad (6)$$

Requiring that such expressions actually characterize a response of the considered plant, as well as the associated output is zero, amounts to ask that:

- During flows, $\dot{x}_{ss}(t, k)$ has to satisfy the two equations:

$$\dot{x}_{ss}(t, k) = \dot{\Pi}(\sigma)w(t, k) + \Pi(\sigma)\dot{w}(t, k),$$

$$\dot{x}_{ss}(t, k) = Ax_{ss}(t, k) + Bu_{ss}(t, k)$$
$$+ Pw(t, k).$$

- At jumps, $x_{ss}^+(t, k)$ has to satisfy the two equations:

$$x_{ss}^+(t_{k+1}, k) = \Pi(0)w^+(t_{k+1}, k),$$

$$x_{ss}^+(t_{k+1}, k) = Ex_{ss}(t_{k+1}, k) + Rw(t_{k+1}, k).$$

- For the output e_{ss} to be identically zero:

$$0 = Cx_{ss}(t, k) + Qw(t, k).$$

Substituting (6) in the above conditions and considering that such relations should hold for all values of w, the following *hybrid regulator equations* are obtained:

$$\dot{\Pi}(\sigma) + \Pi(\sigma)S = A\Pi(\sigma) + B\Gamma(\sigma) + P, \qquad (7a)$$

$$\Pi(0)J = E\Pi(\tau_M) + R, \qquad (7b)$$

$$0 = C\Pi(\sigma) + Q. \qquad (7c)$$

Equations (7) can be shown to be both necessary and sufficient for (6) to solve the output regulation problem under the considered assumptions. Once a solution of (7) is available, the full information regulator simply reduces to the time-varying static feedforward controller

$$u(t, k) = \Gamma(\sigma)w(t, k) \qquad (8)$$

which just provides as input the steady-state input u_{ss} characterized as in (6); in fact, since (5) is incrementally stable (as follows from its asymptotic stability, which was assumed without loss of generality), its output response under the control law (8) must converge to the output response associated to (6).

For later use, note that in the non-hybrid case where $\mathcal{P}$ and $\mathcal{E}$ only flow

$$\dot{w} = Sw, \qquad (9a)$$

$$\dot{x} = Ax + Bu + Pw, \qquad (9b)$$

$$e = Cx + Qw, \qquad (9c)$$

the candidate steady state (6) is replaced by

$$\begin{bmatrix} x_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} \Pi \\ \Gamma \end{bmatrix} w(t), \qquad (10)$$

and (7) reduces to the celebrated *regulator equations* (or *Francis equations*)

$$\Pi S = A\Pi + B\Gamma + P, \qquad (11a)$$

$$0 = C\Pi + Q, \qquad (11b)$$

and, as above, assuming without loss of generality that the plant is asymptotically stable, the full information regulator reduces to the time-invariant static feedforward controller:

$$u(t, k) = \Gamma w(t, k) \qquad (12)$$

The Error Feedback Case

When the exosystem state is not measured, a dynamic compensator of the form

$$\dot{\xi} = F\xi + Ge, \qquad (13a)$$

$$\xi^+ = L\xi, \qquad (13b)$$

$$u = H\xi, \qquad (13c)$$

which is also supposed to flow and jump according to the a priori fixed time domain $\mathcal{T}$ considered for the plant, is introduced, and the corresponding candidate steady-state motion including ξ is

$$\begin{bmatrix} x_{ss}(t, k) \\ \xi_{ss}(t, k) \\ u_{ss}(t, k) \end{bmatrix} = \begin{bmatrix} \Pi(\sigma) \\ \Sigma(\sigma) \\ \Gamma(\sigma) \end{bmatrix} w(t, k). \qquad (14)$$

By following similar steps as above, requiring invariance of such a manifold in the space of (x, ξ, u, w), as well as zero output on it, leads to the conclusion that in addition to (7), the following relations must be satisfied as well:

$$\dot{\Sigma}(\sigma) + \Sigma(\sigma)S = F\Sigma(\sigma), \qquad (15a)$$

$$\Sigma(0)J = L\Sigma(\tau_M). \qquad (15b)$$

$$\Gamma(\sigma) = H\Sigma(\sigma), \qquad (15c)$$

Equations (7) and (15) can be shown to be both necessary and sufficient for (13) to solve the output regulation problem under the considered assumptions and generalize the corresponding conditions for the non-hybrid case where $\mathcal{P}$ and $\mathcal{E}$ only flow (see (9)) and (13) and (14) are replaced by

$$\dot{\xi} = F\xi + Ge, \qquad (16a)$$

$$u = H\xi, \qquad (16b)$$

$$\begin{bmatrix} x_{ss}(t) \\ \xi_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} \Pi \\ \Sigma \\ \Gamma \end{bmatrix} w(t), \qquad (16c)$$

and (15) reduces to

$$\Sigma S = F\Sigma, \qquad (17a)$$

$$\Gamma = H\Sigma. \qquad (17b)$$

Relations (17) are an expression of the *internal model principle*, stating that in order to achieve error feedback regulation, the compensator $\mathcal{C}$ must include a suitable "copy" of the exosystem, namely, (17a) imposes a constraint on the ξ dynamics of $\mathcal{C}$ which, coupled with (17b), ensures that the signal $u_{ss} = \Gamma w$ used in the full information case can be equivalently produced (without measuring w!) as $u_{ss} = H\Sigma\xi$. A similar interpretation can be given to (15), which must be required in addition to (7) in order for (13) to solve the hybrid error feedback output regulation problem.

Key Features in Hybrid vs Classical Output Regulation

While the previous section mainly aimed at showing how the classical theory generalizes in the hybrid case (at least for a special class of hybrid systems), the aim of this section is to point out some of the striking differences between the two

cases. Before proceeding further, and in order to keep focus on the characterization of the steady-state response, it is worth mentioning here that although time-varying systems will be considered, no issue regarding nonuniform stability (like in general nonautonomous systems) arises since the timer σ just ranges in the compact set $[0, \tau_M]$ due to the assumed periodic structure of $\mathcal{T}$ (see also the end of section "Synchronous Jumps").

Comparing the classical and the hybrid output regulator and considering that $\mathcal{P}$ and $\mathcal{E}$ are time invariant, it seems somewhat strange that in the output feedback case the linear time-invariant regulator (16a) and (16b) generalizes to a hybrid linear time-invariant regulator (13), whereas in the full information case the linear time-invariant regulator (12) generalizes to a hybrid linear *time-varying* regulator (8).

One argument in favor of the *time-varying* regulator (8) is based on the following consideration. It is well known that (11) has a unique solution in the case of a square plant ($m = p$) under the *nonresonance condition* between the zeros of $\mathcal{P}$ and the eigenvalues of $\mathcal{E}$, requiring that

$$\text{rank} \begin{bmatrix} A - sI & B \\ C & 0 \end{bmatrix} = n + p, \quad \forall s \in \Lambda(S),$$

where $\Lambda(S)$ denotes the spectrum of S. In such a case, (11) amounts to a system of $nq + pq$ linear equations in $nq + mq$ unknowns (the elements of Π, Γ), which might be expected to be satisfied since $m \geq p$. If one were trying to use the unique constant solution (Π, Γ) of (11) as a solution of (7), clearly (7a) and (7c) would be satisfied, but then (7b) would impose other nq equations on Π which would unlikely be satisfied. For this reason, apparently the additional degree of freedom offered by choosing time dependent Π and Γ might be of help. In fact, it can be shown that if $m = p$ and under a hybrid nonresonance condition (involving $Ee^{A\tau_M}$ and $Je^{S\tau_M}$) between $\mathcal{P}$ and $\mathcal{E}$, (7a) and (7b) have a unique solution for any choice of $\Gamma(\sigma)$, so that the design boils down to satisfying (7c) by choosing $\Gamma(\sigma)$; but is this always possible? In order to answer this nontrivial question, a different path must be followed. While a complete formal analysis can

be performed, the following discussion will be mainly based on showing the simplest examples exhibiting the pathologies of interest.

Consider the system with $\tau_M = 1$ (so that $t_k = k$, for all $k \in \mathbb{Z}$) and

$$\dot{w} = 0, \tag{18a}$$

$$w^+ = -w, \tag{18b}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w, \tag{18c}$$

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2e & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{18d}$$

$$e = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - w. \tag{18e}$$

The unique steady-state solution achieving output regulation can be simply computed. In fact, by (18a) and (18b),

$$w(t, k) = (-1)^k w(0, 0);$$

then, by (18e) it appears that $e_{ss} = 0$, $\forall (t, k) \in \mathcal{T}$ implies

$$x_{2,ss}(t, k) = w(t, k) = (-1)^k w(0, 0),$$

$$\forall (t, k) \in \mathcal{T},$$

which in turn implies that $\dot{x}_{2,ss} = 0$ for all $t \in (k, k+1)$, $k \in \mathbb{Z}$ and the unique steady-state input

$$u_{ss} = 2x_{2,ss} - w.$$

Since (18d) implies that $x_{1,ss}(t_{k+1}, k+1) = x_{2,ss}(t_{k+1}, k) = w(t_{k+1}, k)$ and (18c) implies that $x_{1,ss}(t, k) = -e^{-(t-k)}x_{1,ss}(t_k, k)$, for $t \in (t_k, t_{k+1})$, it follows that

$$x_{1,ss}(t, k) = -e^{-(t-t_k)}w(t_k, k), \quad t \in (t_k, t_{k+1}), \tag{19}$$

which finally is coherent with the jump equation for $x_{2,ss}$ in (18d) since

$$x_{2,ss}(t_{k+1}, k+1) = 2ex_{1,ss}(t_{k+1}, k) + x_{2,ss}(t_{k+1}, k)$$

$$= 2e(-e^{-1})w(t_k, k) + w(t_k, k) \tag{20a}$$

$$= -w(t_k, k) \tag{20b}$$

$$= (-1)^{k+1}w(0, 0). \tag{20c}$$

Before commenting the meaning of the above derived steady-state evolution, it is worth noting that (18) might actually derive from an original system with (18c) replaced by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w, \tag{21}$$

under the preliminary state feedback

$$u = -x_1 + v. \tag{22}$$

Such a feedback renders the subspace $\{x : x_2 = 0\}$ unobservable (when the system only flows) and reveals that the dynamics of x_1 in (18c) is the *flow zero dynamics* of $\mathcal{P}$, that is, the zero dynamics of $\mathcal{P}$ when jumps are inhibited. Having set the stage, several interesting observations can be made now.

The flow zero dynamics samples the exogenous signal w at jumps and then evolves according to its own modes (see (19)). In fact, while in the classical case (10) the state and input at steady state can be expressed as a constant matrix times the current value of w, the real nature of the time dependence of Γ and Π in (6) is linked to this phenomenon of *sampling $w(t_k, k)$* and *propagating along the zero dynamics*. A suitable analysis shows that $\Pi(\sigma)$, $\Gamma(\sigma)$ contain products of matrices with rightmost factor $e^{-S\sigma}$ (which recovers $w(t_k, k) = e^{-S\sigma}w(t, k)$ from the current value $w(t, k)$ of w) and leftmost factor containing the fundamental matrix of the flow zero dynamics. It is worth mentioning that the "motion along the zeros" in the present context is strongly related to the same kind of motions used for perfect tracking in non-hybrid systems. The above insight about the nature of the dependence on σ in (6) also reveals why in the output feedback case (13) such dependence is not needed: the required modes of the flow zero dynamics in

that case are provided by copying them in the compensator dynamics!

An even stronger consequence of the analysis above is a **flow zero dynamics internal model principle**, which essentially states that any output feedback compensator solving the output regulation problem must be able to produce as free responses (during flow) a suitable subset of the natural modes of the flow zero dynamics (and a suitably modified version applies to the feedforward static compensator (8)). It is worth noting that while the classical internal model principle requires exact knowledge of the exosystem modes (which is kind of a mild requirement, especially when the exosystem models references, or constant offsets), the *flow zero dynamics internal model principle* requires the exact knowledge of the modes of the zero dynamics, which typically depends on not precisely known plant parameters; clearly, this fact poses serious questions in view of the achievement of robust regulation.

A final point, also raising serious issues about what can be robustly achieved (and how) in the setting of hybrid output regulation, is the fact that **generically, existence of solutions is not robust to arbitrarily small parameter variations.** In particular, looking again at the computations in (20), it should be clear that the involved functions are all fixed by previous reasonings, whereas satisfaction of (20) crucially depends on exact cancellations of certain coefficients. Any small variations of such coefficients in (18d) imply that the problem admits no steady state yielding zero output. This fact is in sharp contrast with classical regulation, where the nonresonance condition ensures existence of (different) solutions for small parameter variations. It has to be noted, though, that **under additional conditions, robust existence of solutions is guaranteed if the plant is fat**, that is, $m > p$. Using again the previous example, this is the case if an additional input is introduced

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w,$$

since then even a constant (suitably chosen) value of u_1 can be used to ensure that when the time to

jump arrives, the value of x_1 is such to ensure a correct jump for x_2 (remember that since x_1 is unobservable during flows, its motion can be changed as wished if this helps with ensuring that the observable x_2 achieves zero output).

Summary and Future Directions

The investigation of the output regulation problem for hybrid systems is still at a very early stage. While the issues of stabilization of the manifold where regulation is achieved seem to be a relatively better understood topic (possibly drawing from a richer literature on stabilization of hybrid systems), the geometry and design of such manifold appear to involve several much more intricate issues, whose understanding will be crucial in order to achieve more complete solutions.

Already in the very simplified case of linear dynamics and synchronous jumps, the important role played by the whole flow zero dynamics for feasibility (existence of solutions in the nominal parameter values) and by the availability of more inputs than outputs for well posedness (existence of solutions for slightly perturbed parameter values) marks a strong difference with the linear non-hybrid case, where both properties are granted by satisfaction of the nonresonance condition, which only involves the spectrum of the zero dynamics, even for square plants.

While the expected final goal of this investigation should hopefully lead to the design of robust output regulators based on a suitable internal model principle, a deeper understanding of the structure of the steady-state motion achieving regulation, as well as of the effect of additional inputs in shaping it, seems to be an important preliminary step towards such goal.

Cross-References

▶ Hybrid Dynamical Systems, Feedback Control of
▶ Nonlinear Zero Dynamics
▶ Regulation and Tracking of Nonlinear Systems

Recommended Reading

Foundational contributions on classical output regulation are Francis and Wonham (1976), Davison (1976), and Wonham (1985); more recent monographs include Huang (2004), Trentelman et al. (2001), Pavlov et al. (2005), Saberi et al. (2000), and Byrnes et al. (1997). Goebel et al. (2012) provides a solid introduction to a powerful and elegant framework for hybrid systems, including a thorough discussion of stability issues related to those mentioned here. Regulation problems (mainly reference tracking) for classes of hybrid systems with asynchronous jumps are presented in Biemond et al. (2013), Forni et al. (2013a,b), Morarescu and Brogliato (2010), and Galeani et al. (2008, 2012); synchronous jumps (and the ensuing advantages) are considered e.g., Sanfelice et al. (2013). The class of linear systems with synchronous jumps considered in sections "Hybrid Steady State Generation" and "Key Features in Hybrid vs Classical Output Regulation" has been proposed in Marconi and Teel (2010, 2013) and studied in Cox et al. (2011, 2012); the issues related to flow zero dynamics, fat plants and robustness have been discussed in Carnevale et al. (2012a,b, 2013), partly developing remarks contained in Galeani et al. (2008, 2012).

Bibliography

Biemond J, van de Wouw N, Heemels W, Nijmeijer H (2013) Tracking control for hybrid systems with state-triggered jumps. IEEE Trans Autom Control 58(4):876–890

Byrnes CI, Priscoli FD, Isidori A (1997) Output regulation of uncertain nonlinear systems. Birkhäuser, Boston

Carnevale D, Galeani S, Menini L (2012a) Output regulation for a class of linear hybrid systems. Part 1: trajectory generation. In: Conference on decision and control, Maui, pp 6151–6156

Carnevale D, Galeani S, Menini L (2012b) Output regulation for a class of linear hybrid systems. Part 2: stabilization. In: Conference on decision and control, Maui, pp 6157–6162

Carnevale D, Galeani S, Sassano M (2013) Necessary and sufficient conditions for output regulation in a class of hybrid linear systems. In: Conference on decision and control, Florence, pp 2659–2664

Cox N, Teel AR, Marconi L (2011) Hybrid output regulation for minimum phase linear systems. In: American control conference, San Francisco, pp 863–868

Cox N, Marconi L, Teel AR (2012) Hybrid internal models for robust spline tracking. In: Conference on decision and control, Maui, pp 4877–4882

Davison E (1976) The robust control of a servomechanism problem for linear time-invariant multivariable systems. IEEE Trans Autom Control 21(1):25–34

Forni F, Teel AR, Zaccarian L (2013a) Follow the bouncing ball: Global results on tracking and state estimation with impacts. IEEE Trans Autom Control 58(6): 1470–1485

Forni F, Teel AR, Zaccarian L (2013b) Reference mirroring for control with impacts. Daafouz J, Tarbouriech S, Sigalotti M (eds) Hybrid systems with constraints. John Wiley & Sons, Inc., Hoboken, pp 213–260. doi: 10.1002/9781118639856.ch8

Francis B, Wonham W (1976) The internal model principle of control theory. Automatica 12(5):457–465

Galeani S, Menini L, Potini A, Tornambe A (2008) Trajectory tracking for a particle in elliptical billiards. Int J Control 81(2):189–213

Galeani S, Menini L, Potini A (2012) Robust trajectory tracking for a class of hybrid systems: an internal model principle approach. IEEE Trans Autom Control 57(2):344–359

Goebel R, Sanfelice R, Teel A (2012) Hybrid dynamical systems: modeling, stability, and robustness. Princeton University Press, Princeton

Huang J (2004) Nonlinear output regulation: theory and applications, vol 8. Society for Industrial Mathematics, Philadelphia

Marconi L, Teel AR (2010) A note about hybrid linear regulation. In: Conference on decision and control, Atlanta, pp 1540–1545

Marconi L, Teel AR (2013) Internal model principle for linear systems with periodic state jumps. IEEE Trans Autom Control 58(11): 2788–2802

Morarescu I, Brogliato B (2010) Trajectory tracking control of multiconstraint complementarity lagrangian systems. IEEE Trans Autom Control 55(6): 1300–1313

Pavlov AV, Wouw N, Nijmeijer H (2005) Uniform output regulation of nonlinear systems: a convergent dynamics approach. Birkhäuser, Boston

Saberi A, Stoorvogel A, Sannuti P (2000) Control of linear systems with regulation and input constraints. Springer, London

Sanfelice RG, Biemond JJ, van de Wouw N, Heemels W (2013) An embedding approach for the design of state-feedback tracking controllers for references with jumps. Int J Robust Nonlinear Control. doi:10.1002/rnc.2944

Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London

Wonham W (1985) Linear multivariable control: a geometric approach. Applications of mathematics, vol 10, 3rd edn. Springer, New York

O

P

Parallel Robots

Frank C. Park
Robotics Laboratory, Seoul National University,
Seoul, Korea

Abstract

Parallel robots are closed chains consisting of a
fixed and moving platform that are connected by
a set of serial chain legs. Parallel robots typically
possess both actuated and passive joints and may
even be redundantly actuated. Although more
structurally complex and possessing a smaller
workspace, parallel robots are usually designed
to exploit one or more of the natural advantages
they possess over their serial counterparts, e.g.,
higher stiffness, increased positioning accuracy,
and higher speeds and accelerations. In this chap-
ter we provide an overview of the kinematic
and dynamic modeling of parallel robots, a de-
scription of their singularity behavior, and basic
methods developed for their control.

Keywords

Closed kinematic chain; Closed loop mechanism;
Parallel manipulator

Introduction

A parallel robot refers to a kinematic chain in
which a fixed platform and moving platform
are connected to each other by several serial
chains, or legs. The legs, which typically have
the same kinematic structure, are connected to
the fixed and moving platforms at points that are
distributed in a geometrically symmetric fashion.
The Stewart-Gough platform (Fig. 1) is a well-
known example of a parallel robot: each of the
six legs is a *UPS* structure (i.e., consisting of
rigid links serially connected by a universal, pris-
matic, and spherical joint), with the prismatic
joint actuated. Other examples of parallel robots
include the $6 \times RUS$ platform of Fig. 2, the
haptic interface device of Fig. 3, and the eclipse
mechanism of Fig. 4.

Parallel robots can be regarded as a special
class of closed chain mechanisms (i.e., chains
that contain one or more closed loops) and are
purposely designed to exploit the specific ad-
vantages afforded by the closed chain structure,
e.g., for improved stiffness, greater positioning
accuracy, or higher speed. Parallel robots should
be distinguished from two or more cooperating
serial robots that may form closed loops during
execution of a task (e.g., a robotic hand grasping
an object). Some of the fastest velocities and
accelerations recorded by industrial robots have
been achieved by parallel robots, primarily by
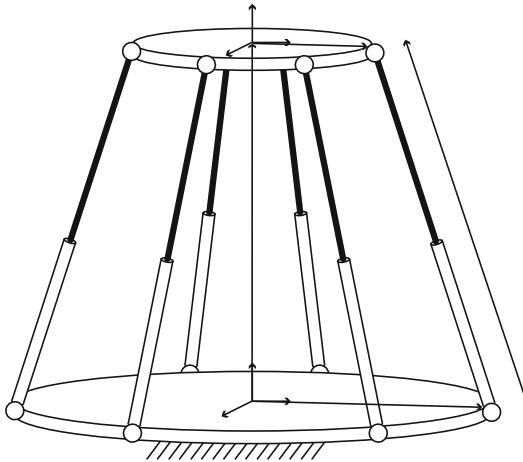
Parallel Robots, Fig. 1 Stewart-Gough platform

placing the actuators on the fixed platform and thereby minimizing the mass of the moving parts.

Many of the model-based techniques developed for the control of traditional serial chain robots are also applicable to a large class of parallel robots. On the other hand, kinematic and dynamic models for parallel robots are inherently more complex. Parallel robots also possess features not found in serial robots, e.g., passive joints, the possibility of redundant actuation, and a diverse range of singularity behavior, that need to be considered when designing a control law. We therefore begin with a brief overview of the kinematic and dynamic modeling of parallel robots before discussing their control.

Modeling

Kinematics

Whereas the kinematic degrees of freedom, or mobility, of a serial chain robot can be obtained as the sum of the degrees of freedom of each of the joints, the situation is somewhat more complex for parallel robots and closed chains in general, since only a subset of the joints can be independently actuated. The mobility of a parallel robot corresponds to the total degrees of freedom of the joints that can be independently actuated. In some cases the number of actuated joint degrees

of freedom may exceed the kinematic degrees of freedom, in which case we say that the robot is redundantly actuated.

A parallel robot with a designated end-effector frame also has a notion of forward and inverse kinematics. While for serial chains the forward kinematics is a well-defined mapping and the inverse kinematics can typically have multiple solutions, for parallel robots the situation is less straightforward. For the Stewart-Gough platform of Fig. 1, in which the leg lengths can be adjusted by actuating the prismatic joints, the inverse kinematics is unique and straightforward to obtain, whereas the forward kinematics will have multiple solutions. For other types of parallel robots in which the legs themselves contain one or more closed loops, both the forward and inverse kinematics can have multiple solutions.

The notion of kinematic singularities for parallel robots is also much more involved than the case for serial robots. Whereas kinematic singularities for serial chain robots are characterized by configurations at which the forward kinematics Jacobian (i.e., the linear mapping relating joint velocities to end-effector frame velocities) becomes singular, for parallel robots and closed chains in general, there exist other notions of singularities not found in serial chains. For example, given a parallel robot with kinematic mobility m – if the parallel robot consists only of one degree-of-freedom joints, this implies that exactly m joints can be actuated – there may exist configurations in which these m joints cannot be independently actuated. Conversely, even if the m actuated joints are each fixed to some value, the parallel robot may fail to be a structure, i.e., some of the links may be able to move.

In the above scenario, choosing a different set of m actuated joints may remedy this situation, in which case such singularities are referred to as actuator singularities. Configurations at which singularity behavior occurs regardless of which joints are actuated are denoted configuration singularities. The final class of singularities are end-effector singularities, which correspond to the usual serial chain notion of kinematic singularity, in which the end-effector loses one or more degrees of freedom of available motion.
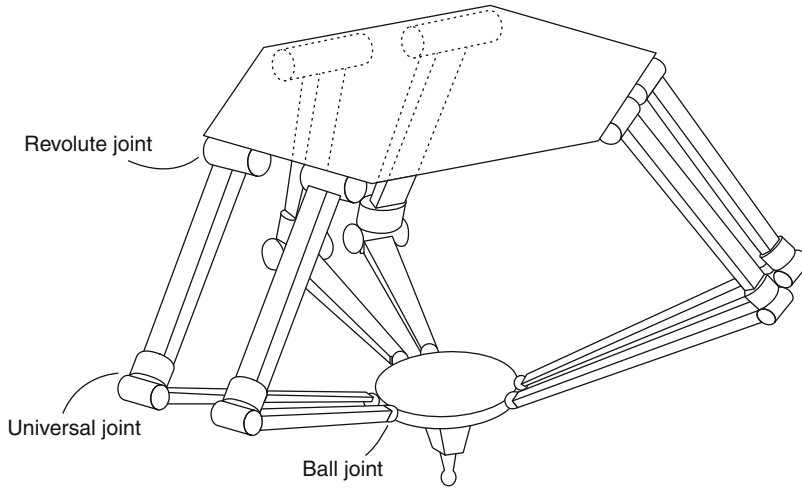
Parallel Robots, Fig. 2 $6 \times RUS$ platform
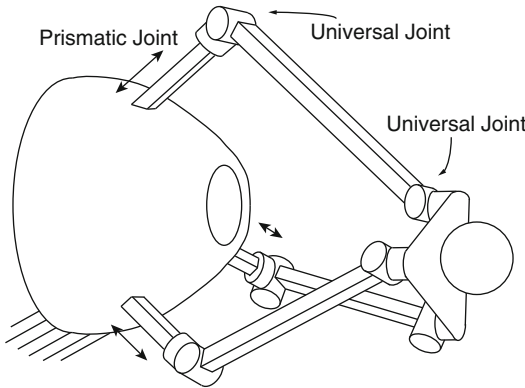
Parallel Robots, Fig. 3 A $3 \times PUU$ haptic interface
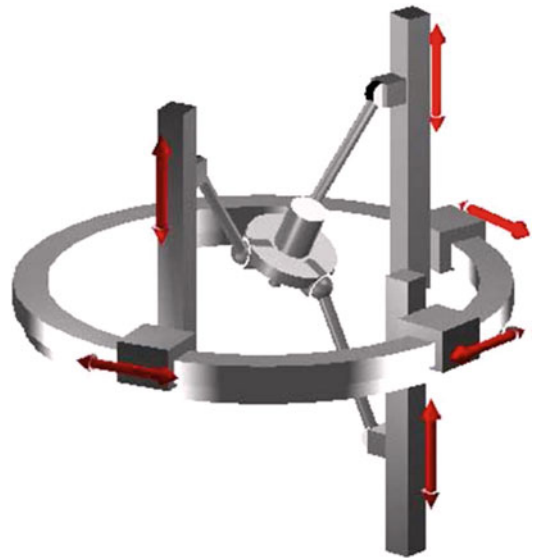
Parallel Robots, Fig. 4 The $3 \times PPRS$ eclipse parallel mechanism

Dynamics

In the case of a parallel robot whose actuated degrees of freedom coincides with its kinematic mobility m, it is possible to choose an independent set of generalized coordinates of dimension m, denoted $q \in \mathbb{R}^m$ and typically identified with the actuated joints, and to express the dynamics in the standard form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau, \quad (1)$$

where $\tau \in \mathbb{R}^m$ denotes the vector of input joint torques, $M(q)$ denotes the $n \times n$ mass matrix, the matrix-vector product $C(q, \dot{q})\dot{q}$ denotes the vector of Coriolis terms, and $G(q) \in \mathbb{R}^m$

denotes the vector of gravitational forces. The structure of the dynamic equations is identical to that for serial chain robots. Also like the case for serial chain robots, the Coriolis matrix term $C(q, \dot{q}) \in \mathbb{R}^{m \times m}$ is not unique, so that one should ensure that the correct $C(q, \dot{q})$ is used in, e.g., any control law whose stability depends on the matrix $\dot{M}(q) - 2C(q, \dot{q})$ being skew-symmetric.

It is also important to keep in mind that the q must satisfy the kinematic constraint equations imposed by the loop closure constraints. That is, if $\theta \in \mathbb{R}^n$ denotes the vector of all joints (both actuated and passive), then $q \in \mathbb{R}^m$, $m \leq n$, will be a subset of θ whose values can only be obtained by solution of the kinematic constraint equations; depending on the nature of the kinematic constraints, one may have to resort to iterative numerical methods.

If the parallel robot is redundantly actuated, then the dynamics are subject to a further set of constraints on the input torques. Letting q_e denote the set of independent generalized coordinates and q_a be the vector of all actuated joints, the vector of actuated joint torques τ_a must then further satisfy $S^T \tau_a = W^T \tau$, where τ denotes the vector of joint torques for an equivalent tree structure system that moves identically to the redundantly actuated parallel robot and W and S are defined, respectively, by

$$S = \frac{\partial \theta}{\partial q_e}, \quad W = \frac{\partial q_a}{\partial q_e}. \quad (2)$$

Compared to the dynamics for serial chain robots, the dynamics for parallel robots is, in general, considerably more complex and computationally involved. The recursive algorithms that are available for computing the inverse and forward dynamics of serial chain robots can also be used to develop similar recursive algorithms for parallel robot dynamics; however, the computations will be considerably more involved and require multiple iterations.

Motion Control

Exactly Actuated Parallel Robots

For parallel robots whose actuated degrees of freedom match the kinematic mobility (this excludes the set of all redundantly actuated parallel robots), most control laws developed for serial chain robots are also applicable. This is not altogether surprising in light of the similarity in the structure of the kinematic and dynamic equations between serial and parallel robots. Control

laws for serial robots are also covered in this handbook, and we refer the reader to ▶ Linear Matrix Inequality Techniques in Optimal Control for the essential details. Here we summarize the most basic control laws and point out any additional computational or other requirements that are needed when applying these laws to parallel robots. Note that other control laws and techniques developed for serial chain robots, e.g., robust, sliding mode, can also be applied with the same additional considerations and requirements outlined below:

1. **Computed torque control**: Computed torque control for parallel robots has the same control law structure as for serial robots, i.e.,

$$\tau = M(q)\left(-K_p e - K_v \dot{e}\right) + \tau_{ff}, \quad (3)$$

where e denotes the tracking error, K_p and K_v are the proportional and derivative feedback gain matrices, and τ_{ff} denotes the feedforward term required to cancel the nonlinear dynamics. Robust versions of computed torque control are also applicable to parallel robots under the same set of conditions, e.g., establishing appropriate bounds on the mass matrix eigenvalues and on the norm of the Coriolis matrix.

2. **Augmented PD control**: The augmented PD control law for serial robots is also applicable to parallel robots, i.e.,

$$\tau = -K_p e - K_v \dot{e} + M(q)\ddot{q}_d + C(q,\dot{q})\dot{q} + G(q), \quad (4)$$

where q_d is the reference trajectory to be tracked and K_p and K_v are the proportional and derivative feedback gains. Asymptotic stability is also established under the same conditions.

3. **Adaptive control**: Because the dynamic equations for parallel robots are also linear in the link mass and inertial parameters, i.e.,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \Phi(q,\dot{q},\ddot{q})p, \quad (5)$$

where p denotes the vector of link mass and inertial parameters, adaptive control laws developed for serial robots can also be used.

4. **Other control methods**: There exist numerous control methods developed for serial robots, e.g., task space or operational space control, sliding mode control, and various nonlinear control techniques; with few exceptions most of these algorithms can also be applied to exactly actuated parallel robots with minimal modification.

Redundantly Actuated Parallel Robots

As described earlier, parallel robots exhibit a much more diverse range of singularity behavior than their serial counterparts, many of which depend on the choice of actuated joints (actuator singularities). One way to eliminate actuator singularities is via redundant actuation, i.e., the total degrees of freedom of the actuated joints exceeds the kinematic mobility of the mechanism. Redundant actuation offers some protection in the event of failed actuators and, when combined with an appropriate control law, offers an effective means of reducing joint backlash, increasing speed and payload and stiffness, controlling compliance through the generation of internal forces, and even improving power efficiency (as an analogy, the human musculoskeletal system is redundantly actuated by antagonistic muscles). Of course, redundant actuation introduces a new set of control challenges, since the control inputs must be designed so as not to conflict with the kinematic constraints inherent in the parallel robot; loosely speaking, the actuated joints can no longer be independently controlled, since the consequences of unintended antagonistic actuation may be catastrophic.

The control of cooperating manipulators (see ▶ Optimal Control and Mechanics) has a long history in robotics, and many of the control techniques developed for such multi-arm systems can also be applied to redundantly actuated parallel robots. One can also apply the control strategies developed for exactly actuated parallel robots to the redundantly actuated case, but modifications

are necessary to account for the different structure of the dynamic equations.

Like all model-based control algorithms, the above control laws are subject to model uncertainties. Whereas in serial chains the effects of model uncertainty simply lead to errors in tracking, for redundantly actuated parallel robots, the consequences can lead to internal forces in addition to end-effector tracking errors. Perhaps the most significant effect of any modeling errors is that, unlike the serial chain case, the kinematic errors can potentially alter the shape of the configuration space (recall that the configuration space will in general be a curved space for closed chains) and also interfere with any PD feedback introduced into the control. The development of control laws that are robust to such modeling errors and disturbances remains an open and ongoing area of research in parallel robot control.

Force Control

Both hybrid force-position control and impedance control are well-known and widely applied concepts in serial robots and can be extended in a straightforward manner to exactly actuated parallel robots. Recall that the basic feature of hybrid force-position control is that the task space is decomposed into force- and position-controlled directions, whereas in impedance control, the goal is have the robot maintain a certain desired spatial stiffness in the task space. Controllers that combine aspects of force-position and impedance control have also been proposed and developed for both serial robots and exactly actuated parallel robots. Modeling errors will cause deviations in both the force- and position-controlled directions – leading to motions in force-controlled directions and forces in position-controlled directions – which can be addressed by, e.g., a switching control strategy.

The problem of force control for redundantly actuated parallel robots, which encompasses both force-position and impedance control, has also received some attention in the literature.

The main difference with the exactly actuated case is that internal forces can now be generated, which requires a more detailed and coordinate-invariant examination of stiffness. Control methods that combine elements of force-position and impedance control for redundantly actuated parallel robots have received only limited attention in the literature.

Cross-References

Recommended Reading

The monograph (Merlet 2006) offers a detailed and comprehensive treatment of all aspects of parallel robots, with a particularly thorough treatment of the kinematics and singularity analysis. Mueller (2008) provides an excellent survey of the dynamics and control of redundantly actuated parallel robots and is based on the preceding work (Mueller 2005). Cheng et al. (2003) examines in detail the dynamic model for redundantly actuated parallel robots and the basic control strategies; Nakamura and Ghodoussi (1989) also examines dynamic models for redundantly actuated parallel robots. Stiffness analysis and control of redundantly actuated parallel robots are addressed in Yi and Freeman (1993), Chakarov (2004), and Fasse and Gosselin (1998). Analysis of specific parallel robots engaged in various control tasks includes Caccavale et al. (2003), Honegger et al. (1997), Kim et al. (2001), and Satya et al. (1995). The basic references on robot control are Murray et al. (1994), Spong et al. (2006), Anderson and Spong (1988), and Ghorbel (1995) focuses on PD control for closed chains.

Bibliography

Anderson RJ, Spong MW (1988) Hybrid impedance control of robotic manipulators. IEEE J Robot Autom 4:549–556

Caccavale F, Siciliano B, Villani L (2003) The Tricept robot: dynamics and impedance control. IEEE/ASME Trans Mechatron 8:263–268

Chakarov D (2004) Study of the antagonistic stiffness of parallel manipulators with actuation redundancy. Mech Mach Theory 39:583–601

Cheng H, Yiu Y-K, Li Z (2003) Dynamics and control of redundantly actuated parallel manipulators. IEEE Trans Mechatron 8(4):483–491

Fasse ED, Gosselin CM (1998) On the spatial impedance control of Gough-Stewart platforms. In: Proceedings of the IEEE international conference on robotics and automation, Leuven, pp 1749–1754

Ghorbel F (1995) Modeling and PD control of closed-chain mechanical systems. In: Proceedings of the IEEE conference on decision and control, New Orleans

Honegger M, Codourey A, Burdet E (1997) Adaptive control of the Hexaglide, a 6-DOF parallel manipulator. In: Proceedings of the IEEE international conference on robotics and automation, Washington, DC, pp 543–548

Kim J, Park FC, Ryu SJ, Kim J, Hwang JC, Park C, Iurascu CC (2001) Design and analysis of a redundantly actuated parallel mechanism for rapid machining. IEEE Trans Robot Autom 17(4):423–434

Merlet JP (1988) Force feedback control of parallel manipulators. In: Proceedings of the IEEE international conference on robotics automation, Philadelphia, pp 1484–1489

Merlet JP (2006) Parallel robots. Springer, Heidelberg

Mueller A (2005) Internal prestress control of redundantly actuated parallel manipulators and its application to backlash avoiding control. IEEE Trans Robot 21(4):668–677

Mueller A (2008) Redundant actuation of parallel manipulators. In: Wu H (ed) Parallel manipulators: towards new applications. I-Tech Publishing, Vienna

Murray R, Li ZX, Sastry S (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton

Nakamura Y, Ghodoussi M (1989) Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators. IEEE Trans Robot Autom 5(3):294–302

Satya SM, Ferreira PM, Spong MW (1995) Hybrid control of a planar 3-DOF parallel manipulator for machining operations. Trans N Am Manuf Res Inst/SME 23:273–280

Spong MW, Hutchinson S, Vidyasagar M (2006) Robot modeling and control. Wiley, New York

Yi B-J, Freeman RA (1993) Geometric analysis of antagonistic stiffness in redundantly actuated parallel mechanisms. J Robot Syst 10:581–603

Particle Filters

Fredrik Gustafsson
Division of Automatic Control, Department of
Electrical Engineering, Linköping University,
Linköping, Sweden

Abstract

The particle filter computes a numeric approximation of the posterior distribution of the state trajectory in nonlinear filtering problems. This is done by generating random state trajectories and assigning a weight to them according to how well they predict the observations. The weights are instrumental in a resampling step, where trajectories are either kept or thrown away. This exposition will focus on explaining the main principles and the main theory in an intuitive way, illustrated with figures from a simple scalar example. A real-time application is used to graphically show how the particle filter solves a nontrivial nonlinear filtering problem.

Keywords

Estimation; Kalman filter; Nonlinear filtering; Sequential Monte Carlo

Introduction

The particle filter computes an arbitrarily good solution to nonlinear filtering problems. The goal in nonlinear filtering is to compute the posterior distribution of the state vector in a dynamic model, given measurements that are related to the state. Bayes rule provides a recursive but computationally intractable solution. Monte Carlo (MC) methods can essentially solve all Bayesian inference problems.

However, for nonlinear filtering, the complexity increases exponentially in time. The MC approach would be to generate a large number of state trajectories (called particles) and their corresponding sequences of predicted measurements and then weighs together the trajectories according to how well the predicted and actual measurement sequences match each other.

With increasing time, the fit is deemed to be poor, since the state space increases exponentially in time. This is usually referred to as the depletion (or degeneracy) problem. The approach in the particle filter is to simulate only one step at the time and then resample the trajectories if needed. For this reason, the particle filter is sometimes referred to as a sequential Monte Carlo method. The resampling step keeps the trajectories that give a good fit, while the bad ones are discarded. The novel idea in the particle filter when it was first published in 1993 was the introduction of this resampling step.

Depletion is still a problem, despite the resampling step. Mitigating depletion has ever since the beginning been the most pressing issue in applied particle filtering. This tutorial will present the basic particle filter algorithm and discuss ways to avoid depletion problems both in general terms and in a simple example.

The particle filter computes an approximation to the Bayes optimal filter, conditioned on a sequence of observations and a nonlinear non-Gaussian system. It is important to note that the PF approximates the posterior distribution of the state trajectory, from which the mean and covariance are easily extracted. In contrast, the extended Kalman filter (EKF) computes the mean and covariance for an approximate dynamical system (linearized with Gaussian noise). The unscented Kalman filter (UKF) likewise also approximates the mean and covariance. Both EKF and UKF can only approximate unimodal (one peak) posterior distributions. There are filter bank approximations, like the interacting multiple model (IMM) algorithm, that can keep track of a given number of modes in the posterior.

P

However, the PF does this in a more natural way.

The Basic Particle Filter

Nonlinear filtering aims at estimating the distribution of a state sequence $x_{1:N} = (x_1, x_2, \ldots, x_N)$ from a sequence of observations $y_{1:N} = (y_1, y_2, \ldots, y_N)$, given a state space model of the form

$$x_{k+1} = f(x_k, v_k) \qquad \text{or} \quad p(x_{k+1}|x_k), \quad \text{(1a)}$$

$$y_k = h(x_k, e_k) \qquad \text{or} \quad p(y_k|x_k). \qquad \text{(1b)}$$

Here, v_k denotes process noise, and e_k is the measurement noise. The stochastic variables v_k, e_k for all k and x_0 are assumed mutually independent, with known distributions p_v, p_e, and p_{x_0}, which are all being part of the model specification.

The particle filter (PF) works with a set of random trajectories. Each trajectory is formed recursively by iteratively simulating the model with some randomness and then updating the likelihood of each trajectory based on the observation. In words, we first evaluate the set of particles at hand by comparing how well they predict the current observation. In this way, the particles are assigned a weight. We keep the particles with large weight and throw away the particles with small weight, using a stochastic resampling procedure. After this step, we get a smaller set of particles, where many particles have several replicas. We then simulate each particle to the next observation time using the dynamical model. After this prediction step, all particles will be unique (because they are based on different realizations of the process noise). Below, the basic algorithm (sometimes called bootstrap PF or sequential importance resampling (SIR) PF) is summarized.

- Define a set of random states (particles) by sampling $x_0^{(i)} \sim p_{x_0}(x_0)$.
- Iterate in $k = 0, 1, \ldots$:
  1 Measurement update: Compute the weight $\omega_k^{(i)} = p_e(y_k - h(x_k^{(i)}))$ and normalize so $\sum_{i=1}^{N} \omega_k^{(i)} = 1$.

  2 Resampling: Resample each particle with probability $\omega_k^{(i)}$.
  3 Time update: Simulate one time step by taking $v_k^{(i)} \sim p_v(v_k)$ and then set $x_{k+1}^{(i)} = f(x_k^{(i)}, v_k^{(i)})$.

The main design parameter here is the number N of particles. A common trick to make the filter more robust is to increase the variance of p_v and p_e above. This is called dithering (or jittering) and is a practical way to get more robust nonlinear filters.

To illustrate some of the aspects and for later reference, a simple example will be introduced.

Example: First-Order Linear Gaussian Model

The Kalman filter (KF) provides the posterior distribution in an analytical form for linear Gaussian models and is thus suitable for evaluations and comparisons. A linear Gaussian model looks like

$$x_{k+1} = F x_k + v_k, \quad v_k \sim \mathrm{N}(0, Q). \quad \text{(2a)}$$

$$y_k = H x_k + e_k, \quad e_k \sim \mathrm{N}(0, R), \quad \text{(2b)}$$

$$x_0 \sim \mathrm{N}(\mu_0, P_0), \qquad\qquad \text{(2c)}$$

We will use the scalar case for the illustrations, and the figures that follow are based on $F = 0.9$, $H = 1$, $Q = 1$, $R = 0.01$, $P_0 = 1$. The particle filter in the scalar case simplifies to the Matlab algorithm in Table 1. Figure 1 compares the sample-based representation of the PF with the Gaussian distribution provided by the KF for the first two time steps. This shows how well the marginal distribution $p(x_k|y_{1:k})$ is approximated by the samples $x_k^{(i)}$ from the PF. A rule of thumb is that 30 samples are needed to approximate a univariate Gaussian distribution. As will be discussed later, the number of samples is effectively only 10 here, which explains the small deviation of the Gaussian functions.

To illustrate the fundamental depletion problem in the PF, the set of trajectories $x_{1:k}^{(i)}$ that

Particle Filters, Table 1 Matlab code for scalar linear Gaussian model

```
% Simulation
y=filter([0 H],[1 -F],[sqrt(P0)*randn(1,1);...
        sqrt(Q)*randn(N-1,1)])+sqrt(R)*randn(N,1);
% Particle filter
x=mu0+sqrt(P0)*randn(N,1);
for k=1:K
    w=exp(-(y(k)-H*x).^2/R);        % Measurement update
    w=w/sum(w);                     % Normalization
    xhat(k)=w'*x;                   % Estimate
    P(k)=w'*(x-xhat(k)).^2;         % Variance
    x=resample(x,w);               % Resampling
    x=F*x+sqrt(Q)*randn(N,1);       % Time update
end
```
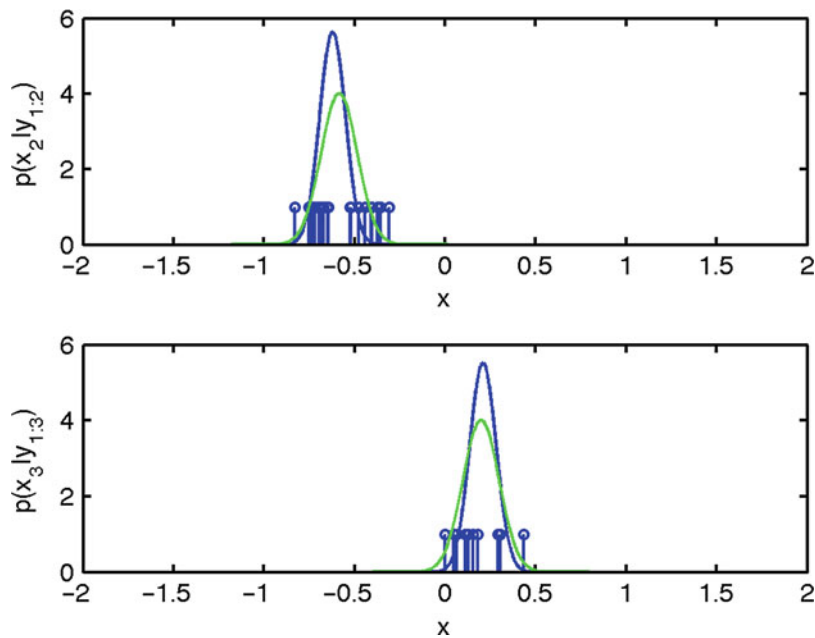

Particle Filters, Fig. 1 Set of samples $\{x_k^{(i)}\}_{i=1:N}$ compared to first a Gaussian approximation of the particles (*blue*) and second to the true posterior distribution provided by the Kalman filter (*green*)

approximates the posterior (smoothing) distribution $p(x_k|y_{1:k})$ is illustrated in Fig. 2. The upper plot shows a case where the trajectories are all the same initially. The behavior in the upper plot is typical for the basic particle filter in cases where the measurements are more informative than the state transition model (small measurement noise, $R < Q$). The lower plot shows a particle filter that is working better, and the modification is explained in the following section.

Proposal Distributions

The time update in the basic PF predicts particles in step 4 according to the dynamic model. The most general derivation of the particle filter allows for sampling from a more general proposal (also called importance) distribution. This proposal distribution can be any function that can be sampled from, and it can depend on both the previous state and the current measurement. From a filtering perspective, it may appear as
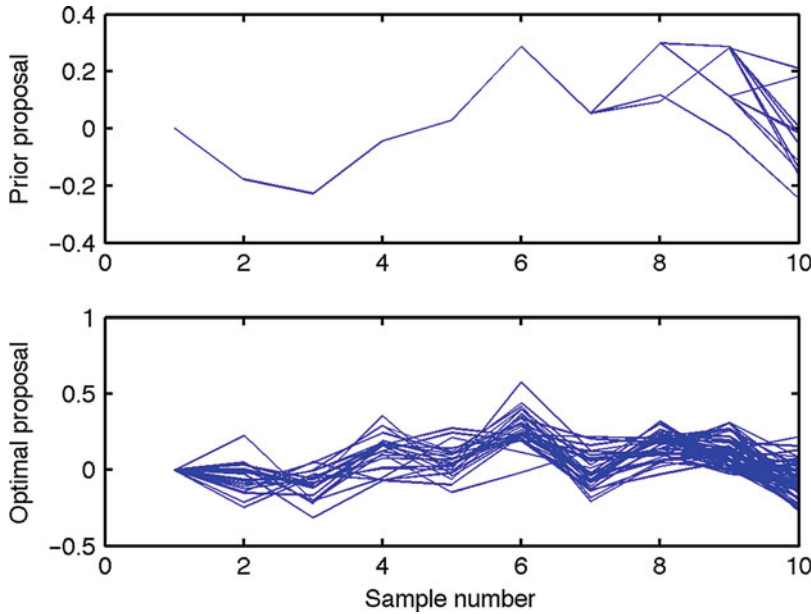
Particle Filters, Fig. 2 Set of trajectories $\{x^{(i)}_{1:10} - x_{1:10}\}_{i=1:N}$ for two different proposal distributions, one bad one (prior) leading to particle depletion and one good

one (likelihood). Note that the smoothing distribution $p(x_k|y_{1:10})$ can be approximated with the set of particles $\{x^{(i)}_k\}_{i=1:N}$ using the marginalization principle

"cheating" to look at the next measurement when doing the time update, but one has to look at a full cycle of the iteration scheme.

If a proposal distribution of the functional form $q(x_k|x_{k-1}, y_k)$ is used, then steps 1 and 3 have to be modified as follows:

1 *Weight update:* Time and measurement updates:

$$w^{(i)}_{k|k-1} \propto w^{(i)}_{k-1|k-1} \frac{p(x^{(i)}_k|x^{(i)}_{k-1})}{q(x^{(i)}_k|x^{(i)}_k, y_k)}, \quad (3a)$$

$$w^{(i)}_{k|k} \propto w^{(i)}_{k|k-1} p(y_k|x^{(i)}_k). \quad (3b)$$

3 *Prediction:* Generate samples from the proposal

$$x^{(i)}_{k+1} \sim q(x_{k+1}|x^{(i)}_k, y_{k+1}) \quad (3c)$$

The most natural proposal distributions are the following:

- The prior $q(x_{k+1}|x^{(i)}_k, y_{k+1}) = p(x_{k+1}|x^{(i)}_k)$, as used in the basic PF.
- The likelihood $q(x_{k+1}|x^{(i)}_k, y_{k+1}) \propto p(y_{k+1}|x_{k+1})$. For the model (2), the proposal becomes $N(y_k/h, R_k/h^2)$.
- The optimal (minimizing weight variance) choice $q(x_{k+1}|x^{(i)}_k, y_{k+1}) \propto p(y_{k+1}|x_{k+1}) p(x_{k+1}|x^{(i)}_k)$. For the model (2), the optimal proposal is provided by one cycle of the Kalman filter, initialized with the particle $x^{(i)}_k$.

The optimal proposal keeps the weights constant, and this would in theory avoid depletion, where depletion is interpreted as excessive weight variance. Figure 2 compares the set of trajectories for the prior and likelihood proposals, respectively. Apparently, the likelihood proposal is to prefer here, since it suffers less from depletion in the particle history. The practical limitation with the last two alternatives is that one has to be able to sample from the likelihood, so in practice there needs to be more measurements than states in the model.
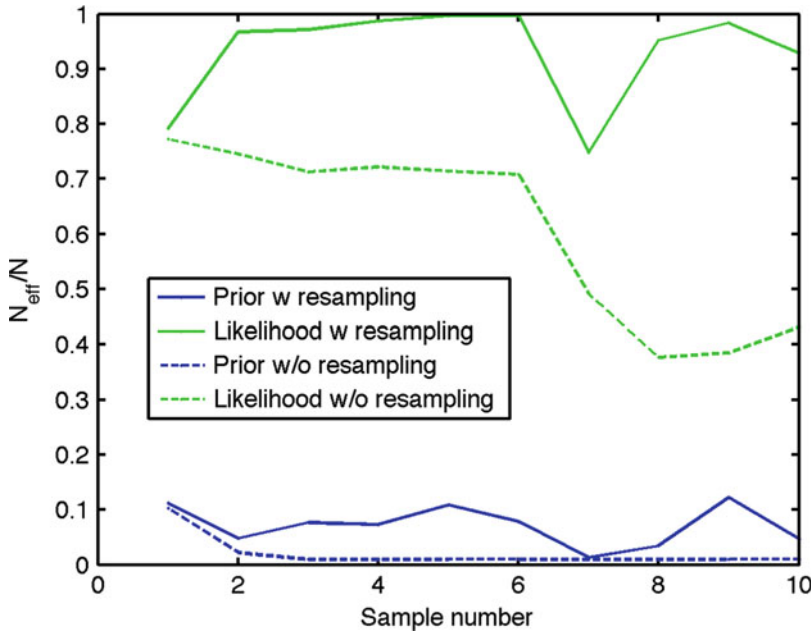
Particle Filters, Fig. 3 The efficient number of particles $N_{\mathrm{eff}}(k)/N$ for prior proposal (*blue*) and likelihood proposal (*green*) ($N = 1{,}000$)

Adaptive Resampling

Resampling is crucial to avoid depletion. Without resampling, all trajectories except for one will get zero weight quite quickly. However, there is no need to resample at every iteration. Actually, resampling increases the weight variance, which is undesired. The question is how to decide if resampling is needed. The efficient number of particles estimated as

$$N_{\mathrm{eff}}(k) = \frac{1}{\sum_{i=1}^{N} \left(w_{k|k}^{(i)} \right)^2}, \qquad (4)$$

is one suitable indicator. If all particles have the same weight $w_{k|k}^{(i)} = 1/N$, then $N_{\mathrm{eff}}(k) = N$. Conversely, if one weight is one and all other zero, then $N_{\mathrm{eff}}(k) = 1$. Thus, N_{eff} can be interpreted as a measure of how many particles that actually contribute to the solution.

Figure 3 shows the evolution of $N_{\mathrm{eff}}(k)$ for prior and likelihood proposals, respectively. With resampling in every iteration, the likelihood pro-

posal performs very well with $N_{\mathrm{eff}}(k) \approx N$, while the prior proposal effectively uses only 10 % of the particles. Thus, $N_{\mathrm{eff}}(k)$ is a good indicator of the quality of the proposal distribution.

In Fig. 1, $N = 100$ so effectively 10 samples are contributing to the Gaussian approximation, which as mentioned before is too small a number to get a good result.

As a comparison, Fig. 3 also shows N_{eff} if resampling is never used, then $N_{\mathrm{eff}}(k)$ normally decreases over time. The likelihood proposal does not decrease as fast as the prior proposal, and for this very short data sequence resampling is really not needed at all.

In summary, resampling increases weight variance and decreases the performance of the filter. On the other hand, without resampling the effective number of particles converges monotonously to only one. So, the idea of adaptive resampling is natural. The key idea is to resample only if the effective number of particles is small. The usual rule of thumb is that resampling is needed if $N_{\mathrm{eff}}(k) < 2N/3$.

Resampling was the main contribution in Gordon et al. (1993) to get a working algorithm.

Marginalization

The posterior distribution approximation provided by the particle filter converges with the number of particles. In theory, the convergence rate is g_k/N, where g_k is a polynomial function in time k. In practice, it appears that the required number of particles increases very quickly with state dimension. Unless a very good proposal distribution is found, the practical limit for the state dimension is around 3–4 as a rough rule of thumb.

For applications with a large number of states, one can in many cases still use the particle filter. The idea is to find a linear Gaussian substructure in the model and then divide the state vector x_k into two parts: x_k^l for the states that appear linearly and x_k^n for the remaining states. Bayes rule provides the factorization

$$p(x_k^l, x_{1:k}^n | y_{1:k}) = p(x_k^l | x_{1:k}^n, y_{1:k}) p(x_{1:k}^n | y_{1:k}) \tag{5}$$

With a linear Gaussian substructure for x_k^l, given the whole trajectory $x_{1:k}^n$, then the Kalman filter applies and provides a Gaussian distribution for the first factor in (5). The second factor is resolved using a marginalization procedure so that the particle filter can be applied.

The bottom line is that each particle is associated with one Kalman filter. The method is called Rao-Blackwellized particle filter, or marginalized particle filter, in literature.

Illustrative Application: Navigation by Map Matching

A nontrivial application where the PF solves a nonlinear filtering problem, where Kalman filter-based approaches would fail, is described in Forssell et al. (2002). The problem is to compute a robust estimate of the position of a car, without using infrastructure such as cellular networks or satellites. The approach is based on measuring wheel speeds on one axle and from that dead reckon a nominal trajectory using standard odometric formulas. A road map is then used as a measurement, to rule out impossible or unlikely maneuvers. There is no numeric measurement y_k in this approach, but the likelihood $p(y_k|x_k)$ in the model (1b) is large when x_k corresponds to a road position and decays quickly to zero outside the road network. Figure 4 illustrates the particle cloud gradually focuses around the true position with time. In particular, the number of modes in the posterior distributions is rather high initially, but decreases over time, in particular after each turn.

The particle filter is sometimes believed to be too computer intensive for real-time applications. As described in Forssell et al. (2002), this demonstrator implemented a particle filter on a pocket computer anno 2001 with $N = 15,000$ particles running in 10 Hz. Thus, the computational complexity of the particle filter should not be overemphasized in practice.

Summary and Future Directions

The particle filter can be seen as a black-box solution to the nonlinear filtering problem, where any nonlinear dynamical model with arbitrary noise distributions can be plugged in. The main tuning parameter is the number of particles, and the PF will work in theory if this number is large enough. In practice, there are many tricks the user has to be aware of to mitigate the curse of dimensionality (depletion) that occurs for large state spaces (more than three) or long time sequences (more than a couple of samples). One engineering trick is dithering, to increase the variance in the involved noise distributions from their nominal values. More theoretical ways to mitigate depletion include clever choices of proposal distributions to sample from and marginalization (to solve a subset of the estimation problem with a Kalman filter).

Current and future research directions include the issues above. A further trend concerns the related smoothing problem, which is
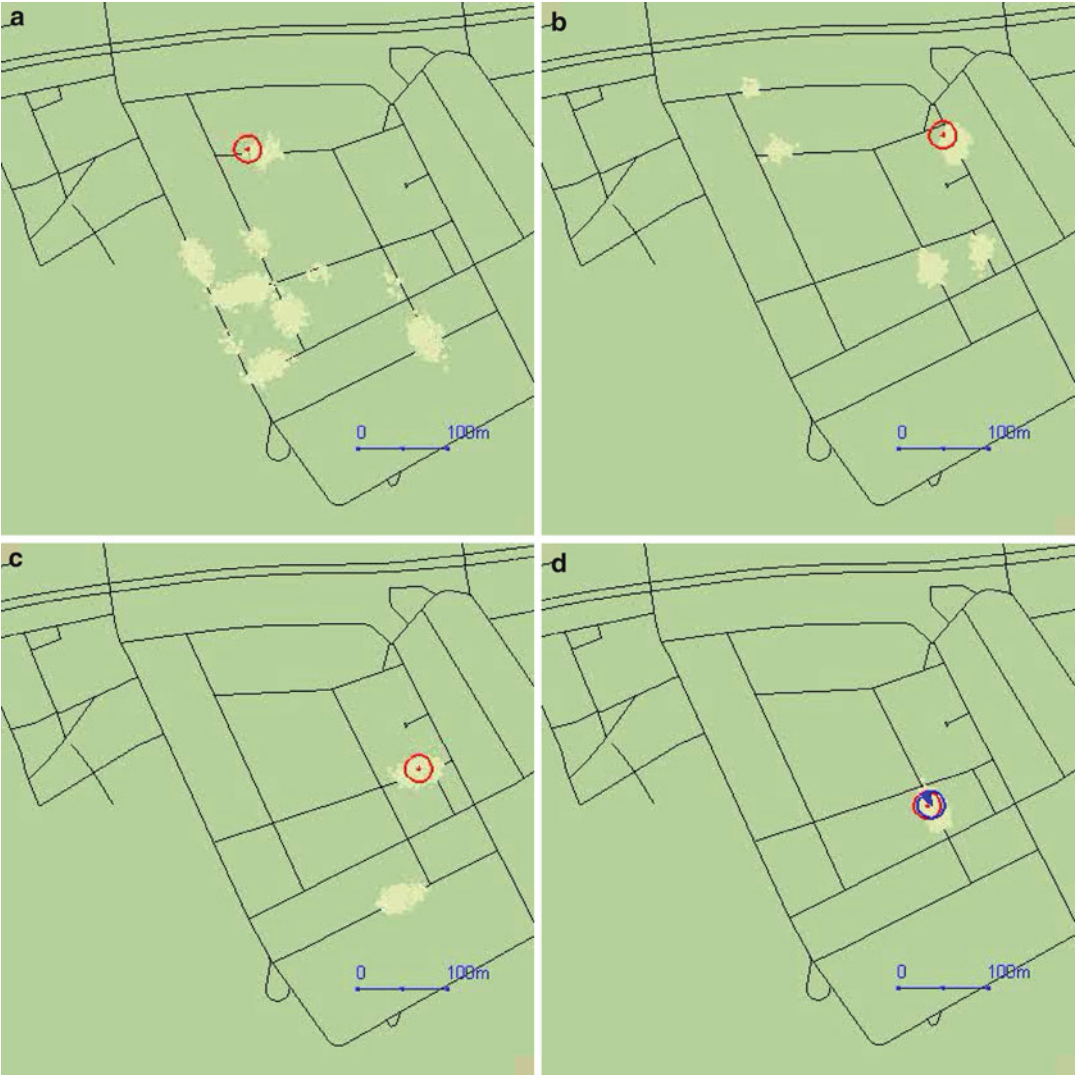
Particle Filters, Fig. 4 Car navigation using wheel speed and a street map. The figures illustrate of how the particle representation of the position posterior distribution of position (course state is now shown) improves over time. After four turns, the posterior is essentially unimodal, and a position marker can be shown. The *circle* denotes GPS position, which is only used for comparison. (**a**) After first turn. (**b**) After second turn. (**c**) After third turn. (**d**) After fourth turn

interesting in itself, but which has turned out to be instrumental in joint state and parameter estimation problems. There are also many attempts to make the particle filter more robust, including ideas of filter banks and invoking a second layer of sampling algorithms to implement the proposal distribution. There is also a trend to use the particle filter as a computational engine for more complex problems, such as the simultaneous localization and mapping (SLAM) problem, and to approximate the probability hypothesis density (PHD) for multi-sensor multi-target tracking. Finally, there are a large number of papers reporting on applications in traditional as well as new disciplines.

Cross-References

Recommended Reading

Particle filtering (PF) as a research area started with the seminal paper (Gordon et al. 1993) and the independent developments in Kitagawa (1996) and Isard and Blake (1998). The state of the art is summarized in the article collection Doucet et al. (2001), the surveys Liu and Chen (1998), Arulampalam et al. (2002), Djuric et al. (2003), Cappé et al. (2007), Gustafsson (2010), and the monograph Ristic et al. (2004).

Bibliography

Arulampalam S, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188

Cappé O, Godsill SJ, Moulines E (2007) An overview of existing methods and recent advances in sequential Monte Carlo. IEEE Proc 95:899

Djuric PM, Kotecha JH, Zhang J, Huang Y, Ghirmai T, Bugallo MF, Miguez J (2003) Particle filtering. IEEE Signal Process Mag 20:19

Doucet A, de Freitas N, Gordon N (eds) (2001) Sequential Monte Carlo methods in practice. Springer, New York

Forssell U, Hall P, Ahlqvist S, Gustafsson F (2002) Novel map-aided positioning system. In: Proceedings of FISITA, Helsinki, number F02-1131

Gordon NJ, Salmond DJ, Smith AFM (1993) A novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc Radar Signal Process 140:107–113

Gustafsson F (2010) Particle filter theory and practice with positioning applications. IEEE Trans Aerosp Electron Mag Part II Tutor 7:53–82

Isard M, Blake A (1998) Condensation – conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28

Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J Comput Graph Stat 5(1):1–25

Liu JS, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. J Am Stat Assoc 93:1032–1044

Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter: particle filters for tracking applications. Artech House, London

Perturbation Analysis of Discrete Event Systems

Yorai Wardi[1] and Christos G. Cassandras[2]
[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
[2]Division of Systems Engineering, Center for Information and Systems Engineering, Boston University, Brookline, MA, USA

Abstract

Perturbation analysis (PA) is a systematic methodology for estimating the sensitivities (gradient) of performance measures in discrete event systems (DES) with respect to various model or control parameters of interest. PA takes advantage of the special structure of DES sample realizations and is based entirely on observable system data. In particular, it does not require knowledge of the stochastic characterizations of the random processes involved and is simple to implement in a nonintrusive manner. PA estimators, therefore, enable implementations for real-time control in addition to off-line optimization. The article presents the main ideas and statistical properties of PA techniques for both DES and recent generalizations to stochastic hybrid systems (SHS), especially for the simplest class of sensitivity estimators known as infinitesimal perturbation analysis (IPA).

Keywords

Gradient estimation; Sample-path techniques; Sensitivity analysis; Stochastic flow models; Queueing networks

Introduction

Sensitivity analysis is an essential component of the system design process in a wide variety of application areas. In essence, it provides quantitative variations of performance metrics resulting from possible perturbations in design set points and, hence, can be used in *optimization and control* as well as provide measures of performance robustness. *Perturbation analysis* (PA) is a systematic technique for computing sample-based sensitivity estimators of performance metrics in discrete event systems (DES) by using the special properties of their sample realizations. The effectiveness of such estimators (e.g., unbiasedness) depends on the characteristics of the DES to which PA is applied and on the specific performance metric of interest. The purpose of this article is to present and explain some of the main ideas and fundamental techniques of PA.

Figure 1 depicts an abstract schematic where the operation of a stochastic system depends on a parameter θ that is chosen from a given set Θ. Let $J(\theta)$ be an expected value performance function of the system, and suppose that $J(\theta) = E\big[L(\theta)\big]$, where $E\big[\,\cdot\,\big]$ denotes expectation and $L(\theta)$ is a sample realization computable from a sample path of the system. In many situations $J(\theta)$ lacks a closed-form expression, and its sample realization, $L(\theta)$, provides the most practical way for its estimation. Applications of sensitivity analysis often concern the effects of perturbations in the parameter θ on the sample realization $L(\theta)$. Denoting such perturbations by $\Delta\theta$, their effects can be characterized by the difference term $L(\theta + \Delta\theta) - L(\theta)$. PA provides such difference terms from the *same* sample path that was used for computing $L(\theta)$. Furthermore, if $\theta \in R^n$ and the function $L(\cdot)$ is differentiable, then PA can compute the gradient term $\nabla L(\theta)$

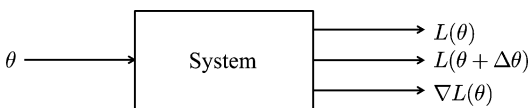**Perturbation Analysis of Discrete Event Systems, Fig. 1** Framework for perturbation analysis (PA)

from the same sample path. These sample path-based sensitivities can be used, under suitable conditions, to estimate the quantities $J(\theta + \Delta\theta) - J(\theta)$ and $\nabla J(\theta)$, respectively.

The PA theory was pioneered by Yu-Chi Ho who led its eventual development by his own group and other researchers over two decades. The early works were motivated by optimal resource management problems in manufacturing and concerned the effects of buffer allocation on throughput in transfer lines (Ho and Cassandras 1983; Ho et al. 1979). Subsequently PA was developed in the setting of queueing networks by virtue of their wide use as models in applications. In this setting, typically θ is a set point parameter affecting service times, inter-arrival times, routing fractions, buffer sizes, and various flow control laws; $J(\theta)$ is an expected value performance metric like average delay, throughput, and loss; and $L(\theta)$ is a sample realization of $J(\theta)$. The special structure of sample paths of queueing networks often yields simple PA algorithms for the difference terms $L(\theta + \Delta\theta) - L(\theta)$, as well as the gradient term $\nabla L(\theta)$, from the common sample path. The PA techniques for computing $L(\theta + \Delta\theta) - L(\theta)$ are collectively referred to as *finite perturbation analysis* (FPA), while those for computing $\nabla L(\theta)$ are called *infinitesimal perturbation analysis* (IPA) (Ho et al. 1983). Much of the development of PA has focused on IPA, rather than FPA, due to its greater simplicity and natural use in optimization, and, hence, it will be the focal point of this article. For comprehensive expositions of PA and its various techniques, please see Ho and Cao (1991), Glasserman (1991), and Cassandras and Lafortune (2008).

The purpose of the IPA gradient, $\nabla L(\theta)$, is to estimate $\nabla J(\theta)$. This, however, is only useful as long as $\nabla L(\theta)$ is an unbiased realization of $\nabla J(\theta)$, namely,

$$E\big[\nabla L(\theta)\big] = \nabla E[L(\theta)] = \nabla J(\theta),$$

and in this case it is said that *IPA is unbiased* (Cao 1985). Since $J(\theta) = E\big[L(\theta)\big]$, unbiasedness means the commutativity of the operators of differentiation with respect to θ and integration (expectation) in the probability space, and this is

closely related to the condition that, w.p.1, the random function $L(\theta)$ is continuous throughout Θ. As a matter of fact, the two conditions are practically synonymous. However, shortly after the emergence of IPA, it became apparent that in many queueing models of interest, $L(\theta)$ is not continuous and, hence, IPA is not unbiased (Heidelberger et al. 1988). Subsequently various techniques to overcome this problem were explored, including the so-called cut and paste of the sample paths and re-parametrization of the underlying probability space via statistical conditioning. For a more comprehensive coverage of such techniques, please see Cassandras and Lafortune (2008) and references therein. These techniques can yield unbiased gradient estimators in principle but often at the expense of prohibitive computing costs. Recently an alternative approach has emerged, based on *stochastic flow models* (SFM) that are comprised of fluid queues (Cassandras et al. 2002). It extends, significantly, the class of models and problems where IPA is unbiased and has the added advantage of yielding very simple gradient estimators.

The following sections of this article present IPA in the general setting of DES, explain the limits of its scope in queueing models, describe alternative PA techniques for extending those limits, present the SFM approach, and conclude with some thoughts on future research directions.

DES Setting for IPA

IPA can be applied to any DES modeled as a stochastic timed automaton, defined in Cassandras and Lafortune (2008) and discussed in ▶ Models for Discrete Event Systems: An Overview. Briefly, a stochastic timed automaton is a sextuple $(\mathcal{E}, \mathcal{X}, \Gamma, p, p_0, G)$, where $\mathcal{E}$ is an event set, $\mathcal{X}$ is a state space, and $\Gamma(x) \subseteq \mathcal{E}$ is the set of feasible events when the state is x, defined for all $x \in \mathcal{X}$. The initial state is drawn from $p_0(x) = P[X_0 = x]$. Subsequently, given that the current state is x, with each feasible event $i \in \Gamma(x)$, we associate a clock value Y_i, which represents the time until event i is to occur. Thus, comparing all such clock values, we identify the

triggering event $E' = \arg\min_{i \in \Gamma(x)}\{Y_i\}$, where $Y^* = \min_{i \in \Gamma(x)}\{Y_i\}$ is the inter-event time (the time elapsed since the last event occurrence). To simplify the notation we define $e' := E'$. Thus, with e' determined, the state transition probabilities $p(x'; x, e')$ are used to specify the next state x'. Finally, the clock values are updated: Y_i is decremented by Y^* for all i (other than the triggering event) which remain feasible in x', while the triggering event (and all other events which are activated upon entering x') is assigned a new lifetime sampled from a distribution G_i. The set $G = \{G_i : i \in \mathcal{E}\}$ defines the stochastic clock structure of the automaton.

Let $T_{\alpha,n}$ denote the nth occurrence time of event $\alpha \in \mathcal{E}$, and let $V_{\alpha,n}$ denote a realization of the lifetime event distribution G_α such that $V_{\alpha,n} = T_{\alpha,n} - T_{\beta,m}$ for some (any) event $\beta \in \mathcal{E}$ and $m \in \{1, 2, \ldots\}$. One can then always write $T_{\alpha,n} = V_{\beta_1,k_1} + \ldots + V_{\beta_s,k_s}$ for some s. Let us now consider a parameter $\theta \in R$ which can only affect one or more of the event lifetime distributions $G_\alpha(x; \theta)$; in particular, θ does not affect the state transition mechanism. The case where $\theta \in R^n$ can be handled in similar ways, but the one-dimensional case permits us to use the derivative notation rather than the gradient symbol, which simplifies the presentation. Viewing lifetimes as functions of θ, $V_{\alpha,k}(\theta)$, it can be shown (under mild technical conditions, see Glasserman (1991)) that

$$\frac{dV_{\alpha,k}}{d\theta} = -\frac{[\partial G_\alpha(x; \theta)/\partial\theta]_{(V_{\alpha,k},\theta)}}{[\partial G_\alpha(x; \theta)/\partial x]_{(V_{\alpha,k},\theta)}}, \quad (1)$$

where the subscript $(V_{\alpha,k}, \theta)$ indicates that the corresponding derivative of G_α is evaluated at the point $(V_{\alpha,k}, \theta)$. This describes how a perturbation in θ *generates* a perturbation in the associated event lifetime $V_{\alpha,k}$. Such a perturbation can now *propagate* through the DES to affect various event occurrence times according to the dynamics prescribed by the stochastic timed automaton. Event time derivatives $dT_{\alpha,n}(\theta)/d\theta$ are given by

$$\frac{dT_{\alpha,n}}{d\theta} = \sum_{\beta,m} \frac{dV_{\beta,m}}{d\theta}\eta(\alpha, n; \beta, m), \quad (2)$$

where $\eta(\alpha, n; \beta, m)$ is a triggering indicator taking values in $\{0, 1\}$ so that $\eta(\alpha, n; \beta, m) = 1$ if the nth occurrence of event α is triggered by the mth occurrence of β, $\eta(\alpha, n; \alpha, n) = 1$ for all $\alpha \in \mathcal{E}$, $\eta(\alpha, n; \beta', m') = 1$ if $\eta(\alpha, n; \beta, m) = 1$ and $\eta(\beta, m; \beta', m') = 1$, and $\eta(\alpha, n; \beta, m) = 0$ otherwise.

This leads to a general-purpose algorithm for evaluating event time derivatives along an observed sample path (see Algorithm 1) of a DES modeled as a stochastic timed automaton. In particular, we define a perturbation accumulator, Δ_α, for every event $\alpha \in \mathcal{E}$. The accumulator Δ_α is updated at event occurrences in two ways: (i) It is incremented by $dV_\alpha/d\theta$ whenever an event α occurs, and (ii) it is coupled to an accumulator Δ_β whenever an event β (possibly $\beta = \alpha$) occurs that activates an event α. No particular stopping condition is specified, since this may vary depending on the problem of interest.

Sample Function Derivatives. Since many sample performance functions $L(\theta)$ of interest can be expressed in terms of event times $T_{\alpha,n}$, we can use (2) and Algorithm 1

Algorithm 1 General-purpose IPA algorithm for stochastic timed automata

1. Initialization
   If event α is feasible at x_0: $\Delta_\alpha := dV_{\alpha,1}/d\theta$
   Else, for all other $\alpha \in \mathcal{E}$: $\Delta_\alpha := 0$
2. Whenever event β is observed
   If event α is activated with new lifetime V_α:

   2.1. Compute $dV_\alpha/d\theta$ through (1)
   2.2. $\Delta_\alpha := \Delta_\beta + dV_\alpha/d\theta$

---

in order to obtain derivatives of the form $dL(\theta)/d\theta$. As an example, a large class of such functions is of the form

$$L_T(\theta) = \int_0^T C(x(t, \theta))dt,$$

where $C(x(t, \theta))$ is a bounded cost associated with operating the system at state $x(t, \theta)$. Then,

$$\frac{dL_T}{d\theta} = \sum_{k=1}^{N(T)} \frac{dT_k}{d\theta}[C(x_{k-1}) - C(x_k)],$$

where $N(T)$ counts the total number of events observed in $[0, T]$ and x_k is the state remaining fixed in any interval (T_k, T_{k+1}) with $T_k = T_{\alpha,n}$ for some $\alpha \in \mathcal{E}$, $n = 1, 2, \ldots$.

Estimation of Performance Measure Derivatives. Using $dL_T(\theta)/d\theta$ from above, we can obtain unbiased estimates of $dJ/d\theta$ if the following condition holds:

$$\frac{dJ(\theta)}{d\theta} = E\left[\frac{dL_T(\theta)}{d\theta}\right].$$

As mentioned earlier, this key condition is closely related to the continuity of the sample performance functions. A discontinuity often is caused by a swap in the order of two events that results from small variations in θ and yields different future state trajectories. However, it is possible that the future state trajectory following the occurrence of the two events is invariant under their order, and in this case the two events are said to *commute*. This commuting condition, defined by Glasserman, was shown to be identical, under broad assumptions, to the continuity of the sample functions $L(\theta)$ and, hence, to the unbiasedness of IPA (Glasserman 1991).

The main ideas discussed in this section will next be illustrated on a simple queue.

Queueing Example

Consider the IPA gradient (derivative) of the expected sojourn time (delay) in a GI/G/1 queue with respect to a real-valued parameter of its arrival process. Assume that the queue is empty at time $t = 0$, and it serves its customers according to the order of their arrivals. Let us denote by a_k, $k = 1, 2, \ldots$, the arrival times, and by s_k, $k = 1, 2, \ldots$, the service times of consecutive customers. Furthermore, we denote by v_k, $k = 1, 2, \ldots$, the kth inter-arrival time, namely, $v_k = a_k - a_{k-1}$, where $a_0 := 0$. Observe that the queue is a stochastic timed automaton as defined earlier,

with event space {arrival, departure}, state space {0, 1, ...} representing the queue's occupancy, and feasible event set {arrival} when $x = 0$ and {arrival, departure} when $x > 0$.

Let $\theta \in R$ is a parameter of the distribution of the inter-arrival times, and, hence, the realizations of v depend on θ in a functional manner. For example, if the arrival process is Poisson and θ is its rate, then a realization of the inter-arrival times has the form $v = -\theta \ln(1 - \omega)$, where $\omega \in [0, 1]$ is a uniform variate. To emphasize the dependence of v on θ, we denote it by $v(\theta)$, while its dependence on ω is implicit. Similarly, the arrival times depend on θ and, hence, are denoted by $a_k(\theta)$, but the service times s_k do not depend on θ. The departure time of the kth customer and its delay depend on θ and are denoted by $d_k(\theta)$ and $D_k(\theta)$, respectively. The forthcoming paragraphs discuss the derivatives of these functions with respect to θ; we use the prime symbol to indicate such derivatives in order to simplify the notation.

Define the sample performance function

$$L_N(\theta) := N^{-1} \sum_{k=1}^{N} D_k(\theta)$$

for a given $N > 0$. Under stability conditions, with probability 1 (w.p.1), $\lim_{N \to \infty} L_N(\theta) = J(\theta)$, where $J(\theta)$ denotes the mean of the delay's stationary distribution. The role of IPA is to estimate $J'(\theta)$ via the sample derivative $L_N'(\theta)$, and this is justified as long as $\lim_{N \to \infty} L_N'(\theta) = J'(\theta)$ w.p.1. In this case, IPA is said to be *strongly consistent*. In contrast, unbiasedness pertains to the performance function $J_N(\theta) := E[L_N(\theta)]$ and means that $E[L_N'(\theta)] = J_N'(\theta)$. The concepts of strong consistency and unbiasedness are closely related except that the latter concerns finite-horizon processes while the former pertains to stationary distributions in steady state. Both concepts have been extensively investigated in recent years: strong consistency in the setting of Markov chains and Markov decision processes Cao (2007) and unbiasedness in the context of stochastic hybrid systems, as will be described in the sequel. We will focus the rest of the discussion on the issue of unbiasedness.

Since $L_N(\theta) = N^{-1} \sum_{k=1}^{N} D_k(\theta)$, its IPA derivative is $L_N'(\theta) = N^{-1} \sum_{k=1}^{N} D_k'(\theta)$, and since $D_k(\theta) = a_k(\theta) - d_k(\theta)$, it follows that $D_k'(\theta) = a_k'(\theta) - d_k'(\theta)$. The last two derivative terms are computable via the following recursive procedures. First, $a_k(\theta) = a_{k-1}(\theta) + v_k(\theta)$, and, hence,

$$a_k'(\theta) = a_{k-1}'(\theta) + v_k'(\theta).$$

The term $v_k'(\theta)$ has to be obtained directly from the realization of v, and this often is possible since such realizations depend functionally on θ; for instance, in the previous example, $v = -\theta \ln(1 - \omega)$ and, hence, $v'(\theta) = -\ln(1 - \omega)$. Next, $d_k(\theta)$ is given by the Lindley equation

$$d_k(\theta) = \max \{a_k(\theta), d_{k-1}(\theta)\} + s_k,$$

and, therefore, denoting by ℓ_k the index of the customer that started the busy period containing customer k, we have that

$$d_k'(\theta) = a_{\ell_k}'(\theta).$$

From these recursive relations it follows that $D_k'(\theta) = 0$ if customer k starts a busy period and $D_k'(\theta) = -\sum_{i=\ell_k+1}^{k} v_i'(\theta)$ if customer k does not start a busy period.

These equations shed light on the structure of IPA in a general class of queueing networks. First there is the *perturbation generation*, namely, the sampling of derivative (gradient) terms directly from the sample sequence of variates (ω) which defines the sample path; that was $v'(\theta)$ in the above example. These terms drive the recursive equations that yield the IPA derivatives. The recursive equations often are based on the tracking of certain events such as the start of busy periods or idle periods, and the process of tracking the derivatives through them is referred to as *perturbation propagation*. In the above example it is obvious how the perturbation propagation tracks the busy periods at the queue. Furthermore, in a network setting, the perturbations can propagate from one queue to the next in a natural fashion. For instance, suppose that customers departing

$v_{1,k}(\theta)$

$v_{2,m}$
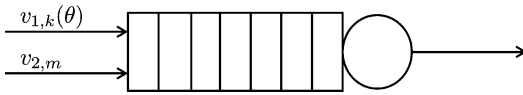
Perturbation Analysis of Discrete Event Systems, Fig. 2 Queue with two customer classes

from the queue analyzed above enter a second queue. Then the derivative terms $d_k'(\theta)$ of the upstream queue act as the derivative terms $a_k'(\theta)$ of the downstream queue. This structure of perturbations' generation and propagation through the tracking of busy periods and other events often yields simple recursive algorithms for computing the IPA derivatives.

Concerning the issue of unbiasedness, it is clear that in the above example, the sample function $L_N(\theta)$ is continuous in θ and, hence, IPA is unbiased. However, in many systems of interest the IPA, derivative is biased. For example, consider the two-input, single-server queue shown in Fig. 2, where customers are served according to their arrival order regardless of source. Suppose that θ is a parameter of the upper arrival process, but not of the lower arrival process, and denote the respective inter-arrival times of the input streams by $v_{1,k}(\theta)$, $k = 1, 2, \ldots$, and $v_{2,m}$ $m = 1, 2, \ldots$, as indicated in the figure. Furthermore, let $d_{1,k}(\theta)$ denote the departure time from the queue of the kth customer that came from the upper source, and let $d_{2,m}(\theta)$ denote the departure time from the queue of the mth customer that came from the lower source. Similarly, let $D_{1,k}(\theta)$ and $D_{2,m}(\theta)$ be the delays of the kth customer from the upper source and the mth customer from the lower source, respectively. Lastly, in analogy with the previous example, consider the sample performance functions $L_{1,N}(\theta) := N^{-1} \sum_{k=1}^{N} D_{1,k}(\theta)$ and $L_{2,N}(\theta) := N^{-1} \sum_{m=1}^{N} D_{2,m}(\theta)$.

The IPA derivatives $L_{1,N}'(\theta)$ and $L_{2,N}'(\theta)$ have quite similar expressions to those derived earlier, but they are not unbiased. To see this point, suppose that $v_{1,k}(\cdot)$ is a monotone increasing function of θ, and consider the functions $a_{1,k}(\theta)$, $d_{1,k}(\theta)$, and $d_{2,k}(\theta)$ for a common sample path. Suppose that at some point $\bar{\theta}$ the order of arrivals of the nth customer from the upper source

and the mth customer from the lower source is swapped. Then the service order of these customers will be swapped as well, inducing discontinuities in $d_{1,k}(\theta)$ and $d_{2,m}(\theta)$ at the point $\theta = \bar{\theta}$. Consequently, the sample performance functions $L_{1,N}(\theta)$ and $L_{2,N}(\theta)$ also will be discontinuous at $\theta = \bar{\theta}$, and hence, their IPA derivatives are biased. Furthermore, if the queue is a part of a network and its output process directs customers to other queues, then the discontinuities in the various traffic processes will propagate downstream.

The causes of biasedness in queueing networks include multiple customer classes, non-Markovian routing, and loss (spillover) due to finite buffers. This leaves out a limited class of networks where IPA can be unbiased and, hence, useful in applications. The following sections describe various approaches to overcome this problem.

IPA Extensions

When IPA fails (because the commuting condition is violated or a sample function exhibits discontinuities in θ), one can still use the PA approach to derive unbiased performance sensitivity estimates. There are two ways to accomplish this: (i) by modifying the stochastic timed automaton model so that IPA is "made to work" and (ii) by paying the price of more information collected from the observed sample path, in which case, the same essential PA philosophy can lead to unbiased and strongly consistent estimators, but these are no longer as simple as IPA ones. Regarding (i), the main idea here is that there may be more than one way to construct (statistically equivalent) sample paths of a stochastic DES, and while one way leads to discontinuous sample functions $L(\theta)$, another does not; a variety of such ways is provided in Cassandras and Lafortune (2008). Regarding (ii), the methodology of *smoothed perturbation analysis* (SPA) (Gong and Ho 1987) provides a generalization of IPA in which more information is extracted from a DES sample path in order to gain some knowledge about the magnitude of jumps in $L(\theta)$.

P

The main idea of SPA lies in the "smoothing property" of conditional expectation. If we are willing to extract information from a sample path and denote it by $\mathcal{Z}$, called the *characterization* of the sample path, then we can evaluate, not just the sample function $L(\theta)$, but also the conditional expectation $E\big[L(\theta) \mid \mathcal{Z}\big]$ (provided we have some distributional knowledge based on which this expectation can be evaluated). This can result in a much smoother function of θ than $L(\theta)$. Thus, starting with the condition for an IPA estimator to be unbiased,

$$\nabla J(\theta) = E\big[\nabla L(\theta)\big],$$

we rewrite the left-hand side above as shown below, replacing $J(\theta) = E\big[(L(\theta)\big]$ by the expectation of a conditional expectation:

$$\nabla J(\theta) = \nabla E\big[L(\theta)\big] = \nabla E\big[E[L(\theta) \mid \mathcal{Z}]\big], \tag{3}$$

where the inner expectation is a conditional one and the conditioning is on the characterization $\mathcal{Z}$. Treating $E\big[L(\theta) \mid \mathcal{Z}\big]$ as the new sample function, we expect it to be "smoother" than $L(\theta)$, and, in particular, continuous in θ. Then, under some additional conditions (comparable to those made in the development of IPA) the interchange of differentiation and expectation in (3) can be justified:

$$\nabla J(\theta) = E\big[\nabla E[L(\theta) \mid \mathcal{Z}]\big].$$

Letting $L_{\mathcal{Z}}(\theta) := E\big[L(\theta) \mid \mathcal{Z}\big]$, the SPA estimator of $\nabla J(\theta)$ is

$$[\nabla J(\theta)]_{\text{SPA}} = \nabla L_{\mathcal{Z}}(\theta). \tag{4}$$

Naturally, the idea is to minimize the amount of added information represented by $\mathcal{Z}$, since this incurs added costs we would like to avoid. The choice of the characterization $\mathcal{Z}$ generally depends on the sample function considered and the system under study.

A number of other extensions to IPA have also been developed (see Cassandras and Lafortune 2008). It is also worth mentioning that the PA approach can be applied to a parameter θ taking values from a finite set $\Theta = \{\theta_0, \theta_1, \ldots, \theta_M\}$. The theory of *concurrent estimation* and *sample path constructability* provides techniques to estimate performance measures of the DES through the process of constructing sample paths under each of $\theta_0, \theta_1, \ldots, \theta_M$; for details see Cassandras and Lafortune (2008).

SFM Framework for IPA

The stochastic flow model (SFM) framework essentially consists of fluid queues which forego the notion of the individual customer and focus instead on the aggregate flow. In such a fluid queue, traffic and service processes are characterized by instantaneous flow rates as opposed to the arrival, departure, and service times of discrete customers. The SFM qualifies as a *stochastic hybrid system* with bilayer dynamics: discrete event dynamics at the upper layer and time-driven dynamics at the lower layer. The discrete events are associated with abrupt (discontinuous) changes in traffic-flow processes, such as the boundaries of busy periods at the queues. In contrast, the time-driven dynamics describe the continuous evolution of flow rates between successive discrete events, usually by differential equations or explicit functional terms. Performance metrics that are natural to SFMs typically reflect quantitative measures of flow rates, like average throughput, buffer workload, and loss.

Due to the smoothing effects of SFMs, they appear to provide a far more natural setting for IPA than their analogous discrete queueing counterparts. Furthermore, their IPA gradients often are computable via extremely simple algorithms that are based entirely on the observed sample path. Consequently SFMs could, in principle, be implemented on the sample paths generated by an actual system rather than simulations thereof and thus be used in real-time optimization.

All of this next will be explained via a concrete example of a queue which, though simple, captures the salient features of the SFM setting for IPA. For a more comprehensive discussion, please see Cassandras et al. (2010), Wardi et al. (2010), and Yao and Cassandras (2013).
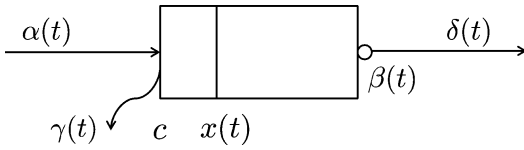
Perturbation Analysis of Discrete Event Systems, Fig. 3 Basic SFM

Consider the fluid queue depicted in Fig. 3 whose input and output flow rate processes are denoted, respectively, by $\alpha(t)$ and $\delta(t)$. The output flow process depends on the input flow process via the action of the server as well as the buffer size. The server is characterized by an instantaneous processing rate, denoted by $\beta(t)$, and the buffer size, namely, the maximum amount of fluid the buffer can hold, is denoted by c. Fluid overflow occurs when the inflow (arrival) rate exceeds the service rate while the buffer is full, and the overflow (loss) rate is denoted by $\gamma(t)$.

Suppose that $\alpha(t)$ and $\beta(t)$ are random functions defined on a suitable probability space, and assume that they are piecewise continuous and of bounded variation w.p.1. In order to describe their functional relations to $\delta(t)$ and $\gamma(t)$, we define the state variable to be the amount of fluid in the buffer (workload) and denote it by $x(t)$. The dynamics of the system evolve according to the following one-sided differential equation,

$$\frac{dx}{dt^+} = \begin{cases} 0, & \text{if } x(t) = 0 \text{ and } \alpha(t) \leq \beta(t) \\ 0, & \text{if } x(t) = c \text{ and } \alpha(t) \geq \beta(t) \\ \alpha(t) - \beta(t), & \text{otherwise}, \end{cases}$$

and $\delta(t)$ and $\gamma(t)$ are related to them via

$$\delta(t) = \begin{cases} \beta(t), & \text{if } x(t) > 0 \\ \alpha(t), & \text{if } x(t) = 0, \end{cases}$$

and

$$\gamma(t) = \begin{cases} \alpha(t) - \beta(t), & \text{if } x(t) = c \\ 0, & \text{if } x(t) < c. \end{cases}$$

Network arrangements of such fluid queues, with specified routing and control schemes, provide a rich class of SFMs.

Let $\theta \in R$ be a parameter of the inflow rate, the service rate, or a network control law; for instance, θ can be the *on* time of the flow from an off/on source, a uniform rate of the server, or the threshold level in a threshold-based flow control. Then the aforementioned traffic processes are functions of θ and t and, hence, are denoted by $\alpha(\theta; t)$, $\beta(\theta; t)$, $x(\theta; t)$, etc. Fix the parameter θ, and consider the evolution of the system over a given time horizon $[0, T]$. Performance measures of interest in applications include the average loss rate over the horizon $[0, T]$ and the average workload there which is related to the delay by Little's Law. Related to them are the sample performance functions $L_{\gamma,T}(\theta) := \int_0^T \gamma(\theta, t)dt$ and $L_{x,T}(\theta) := \int_0^T x(\theta, t)dt$; the former is called the *loss volume* and the latter, the *cumulative workload*.

To illustrate the forms of their IPA derivatives, consider the basic SFM shown in Fig. 3, and let θ be its buffer size, namely, $\theta = c$. We say that a busy period of the queue is *lossy* if it incurs some loss at any amount. Let us denote by N_T the number of lossy busy periods in the horizon $[0, T]$. Then (see Cassandras et al. 2002), the IPA derivative of the loss volume has the following form:

$$L'_{\gamma,T}(\theta) = -N_T,$$

where again we use the prime symbol to denote derivative with respect to θ. This formula amounts to a counting process and indeed it is very simple. As an example, Fig. 4 depicts a typical state trajectory derived from a sample path. It is readily seen that the first busy period is lossy while the second one is not, and therefore, $L'_{\gamma,T}(\theta) = -1$.

Concerning the cumulative workload, suppose that the queue has M lossy busy periods in the time interval $[0, T]$, and let us enumerate them by the counter $m = 1, \ldots, M$. Moreover, denote by u_m the first time the buffer becomes full in its mth lossy busy period and by v_m, the end-time of that busy period. Then (see Cassandras et al. 2002),

$$L'_{x,T}(\theta) = \sum_{m=1}^{M} (v_m - u_m).$$

In the example provided by Fig. 4, $L'_{x,T}(\theta) = v_1 - u_1$.
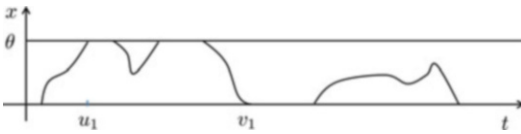
Perturbation Analysis of Discrete Event Systems, Fig. 4 State trajectory of the SFM

These equations for the IPA derivatives not only are very simple, but require no knowledge of the specific form of the processes $\{\alpha(t)\}$ or $\{\beta(t)\}$, their realizations, or underlying probability law. They depend only on limited information which is directly observed from the sample path and, hence, are said to be *nonparametric* or *model free*. Furthermore, they have been shown to possess considerable robustness to modeling variations that do not cause significant alterations of the busy periods of the queue. A case of interest is when the SFM formalism is used as an abstraction of a queue. Then the above IPA formulas that were derived from an analysis of the SFM can be successfully applied to sample paths that are generated from the discrete queue. This is in contrast to the IPA formulas that are derived from the discrete queue, which generally are highly biased.

All of these properties of IPA in the SFM setting, including its unbiasedness, simplicity, the nonparametric nature of its algorithms, and its robustness to model variations, have had extensions to SFM networks and systems beyond the basic model (see Cassandras et al. 2010; Wardi et al. 2010; Yao and Cassandras 2013). As mentioned above, this suggests the potential application of IPA not only in system optimization via off-line simulation but also in real-time control where the sample paths are generated from the actual system.

Summary and Future Directions

In the past 10 years, the focus of research on IPA has shifted from the setting of queueing systems to the framework of SFMs. The main reason for this shift is that IPA yields unbiased gradient estimators for a considerably richer class of networks and performance functions in the SFM setting than for their analogous queueing models. Furthermore, the algorithms for computing the IPA gradients often are nonparametric, robust to modeling variations, and very simple to compute, and, hence, they hold out promise of implementations in real-time control in addition to off-line optimization.

In analogy with the extension of the scope of IPA from queueing systems to stochastic timed automata, the SFM framework has been extended to *stochastic hybrid systems* (SHS), defined in Cassandras et al. (2010). In such systems, including SFMs, the functional description of the time-driven dynamics is changed according to the occurrence of specific events. However, whereas in SFMs these dynamics are described by explicit functional relations, in SHS they are expressed by differential equations. The aforementioned, appealing properties of IPA gradients in the SFM setting appear to have extensions to the wider context of SHS.

Future directions in the use of IPA are expected to focus on the control of high-speed large-volume networks and, more generally, the control of stochastic hybrid systems.

Cross-References

▶ Models for Discrete Event Systems: An Overview
▶ Perturbation Analysis of Steady-State Performance and Sensitivity-Based Optimization

Bibliography

Cao X-R (1985) Convergence of parameter sensitivity estimates in a stochastic experiment. IEEE Trans Autom Control 30:834–843

Cao X-R (2007) Stochastic learning and optimization: a sensitivity-based approach. Springer, Boston

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems, 2nd edn. Springer, New York

Cassandras CG, Wardi Y, Melamed B, Sun G, Panayiotou CG (2002) Perturbation analysis for on-line control and optimization of stochastic fluid models. IEEE Trans Autom Control 47: 1234–1248

Cassandras CG, Wardi Y, Panayiotou CG, Yao C (2010) Perturbation analysis and optimization of stochastic hybrid systems. Eur J Control 16:642–664

Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer, Boston

Gong WB, Ho YC (1987) Smoothed perturbation analysis of discrete event systems. IEEE Trans Autom Control 32:858–866

Heidelberger P, Cao X-R, Zazanis M, Suri, R (1988) Convergence properties of infinitesimal perturbation analysis estimates. Manag Sci 34:1281–1302

Ho YC, Cao, X-R (1991) Perturbation analysis of discrete event dynamical systems. Kluwer, Boston

Ho YC, Cassandras CG (1983) A new approach to the analysis of discrete event dynamic systems. Automatica 19:149–167

Ho YC, Eyler MA, Chien DT (1979) A gradient technique for general buffer storage design in a serial production line. Int J Prod Res 17:557–580

Ho YC, Cao X-R, Cassandras CG (1983) Infinitesimal and finite perturbation analysis for queueing networks. Automatica 19:439–445

Wardi Y, Adams R, Melamed B (2010) A unified approach to infinitesimal perturbation analysis in stochastic flow models: the single-stage case. IEEE Trans Autom Control 55:89–103

Yao C, Cassandras CG (2013) Perturbation analysis and optimization of multiclass multiobjective stochastic flow models. Discret Event Dyn Syst 21:219–256

Perturbation Analysis of Steady-State Performance and Sensitivity-Based Optimization

Xi-Ren Cao
Department of Finance and Department of Automation, Shanghai Jiao Tong University, Shanghai, China
Institute of Advanced Study, Hong Kong University of Science and Technology, Hong Kong, China

Abstract

We introduce the theories and methodologies that utilize the special features of discrete event dynamic systems (DEDSs) for perturbation analysis (PA) and optimization of steady-state performance. Such theories and methodologies usually take different perspectives from the traditional optimization approaches and therefore may lead to new insights and efficient algorithms.

The topic discussed includes the gradient-based optimization for systems with continuous parameters and the direct-comparison-based optimization for systems with discrete policies, which is an alternative to dynamic programming and may apply when the latter fails. Furthermore, these new insights can also be applied to continuous-time and continuous-state dynamic systems, leading to a new paradigm of optimal control.

Keywords

Gradient estimation; Sample-path techniques; Sensitivity analysis; Queueing networks

Introduction

In this chapter, we introduce the theories and methodologies that utilize the special features of discrete event dynamic systems (DEDSs) for perturbation analysis (PA) and optimization of steady-state performance. Such theories and methodologies usually take different perspectives from the traditional optimization approaches and therefore may lead to new insights and efficient algorithms. Furthermore, these new insights can also be applied to continuous-time and continuous state dynamic systems, leading to a new paradigm of optimal control.

As discussed in ▶ Perturbation Analysis of Discrete Event Systems, perturbation analysis (PA) can be applied to both performance in finite-period and steady-state performance. This chapter will mainly focus on the latter and a related topic, the sensitivity-based optimization of steady-state performance of stochastic discrete event dynamic systems.

Gradient-Based Approaches

Basic Ideas

The gradient-based performance optimization of discrete event dynamic systems (DEDSs) consists of three steps:

P

1. Developing efficient algorithms to estimate the performance gradients using the special features of a DEDS
2. Studying the properties of the gradient estimates, including investigating whether they are unbiased and/or strongly consistent
3. With the gradient estimates, developing efficient optimization algorithms

Steps 1 and 2 are referred to as PA, and Step 3 is usually done together with standard gradient-based optimization approaches, such as hill-climbing type of approaches and stochastic approximation approaches such as the Robbins-Monro algorithm (Robbins and Monro 1951).

Our focus here is on PA for steady-state performance. The main principle for estimating the gradients of steady-state performance is decomposition. In a DEDS, a small change in the value of a system parameter induces a series of changes on the system's sample path; each of such changes is called a *perturbation*. A single perturbation alone will affect the sample path and therefore affect the system performance. Such an effect is typically small, and therefore, the linear superposition usually holds. Thus, the effect of a small parameter change on the steady-state performance can be determined by summing up the effects of all the perturbations induced (or generated) by the parameter change. This principle is illustrated by Fig. 1, and it applies to many different systems with different performance criteria. Following the principle, efficient algorithms can be developed, and their strong consistency can be

proved Cao (2007). Its application to queueing systems and Markov systems will be discussed in the following two subsections.

Queueing Systems

The gradient-based approach for DEDSs starts with queueing systems, known as infinitesimal perturbation analysis (IPA), or simply PA, ▶ Perturbation Analysis of Discrete Event Systems. Queueing systems are widely used as a model for many DEDSs in literature and have very unique structural features, and PA utilizes such special features to develop fast algorithms to estimate the performance gradients.

We first give a brief explanation for the simple rules of PA of queueing systems (Ho and Cao 1983, 1991). Consider a closed Jackson network with M servers and N customers. The service times of customers at server i are exponentially distributed with service rate μ_i, $i = 1, 2, \cdots, M$. After a customer completes its service at server i, it goes to server j with a routing probability q_{ij}, $i, j = 1, 2, \cdots, M$. The service discipline is first come first served. The number of customers at server i is denoted as n_i, and the system state is denoted as $\boldsymbol{n} = (n_1, n_2, \cdots, n_M)$. The state process is $\boldsymbol{n}(t) = (n_1(t), n_2(t), \cdots, n_M(t))$ with $n_i(t)$ being the number of customers at server i at time t, $i = 1, 2, \cdots, M$, $t \geq 0$. Define T_l as the lth state transition time of the process $\boldsymbol{n}(t)$.
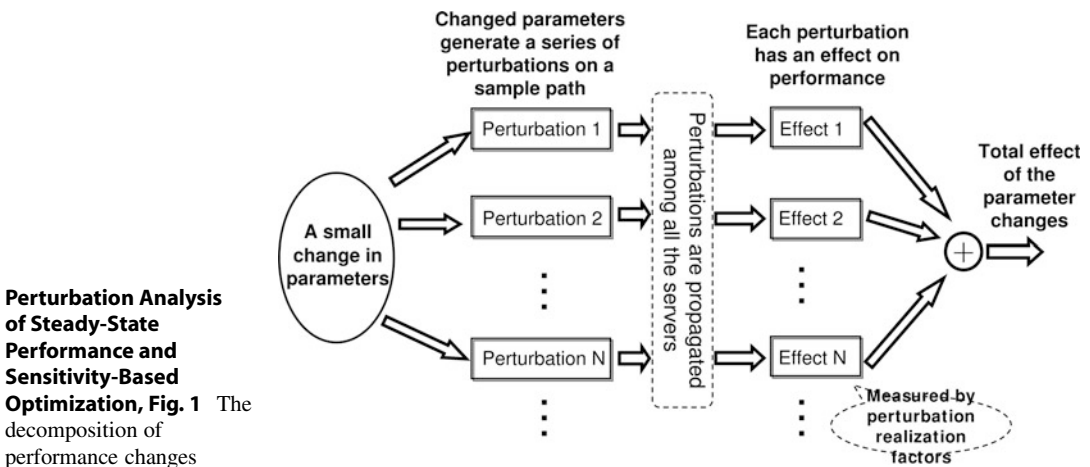
Perturbation Analysis of Steady-State Performance and Sensitivity-Based Optimization, Fig. 1 The decomposition of performance changes

Perturbation Analysis of Steady-State Performance and Sensitivity-Based Optimization, Fig. 2 Perturbation generation and propagation in a queueing network
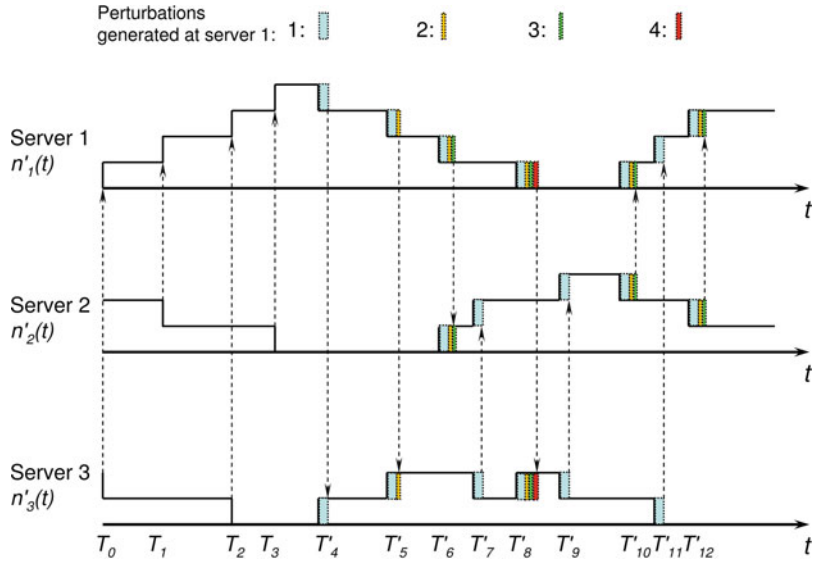
Figure 2 illustrates an example of a sample path where each stair-style line represents a trajectory of the number of customers at one server, and the customer transitions among servers are indicated by dotted arrows.

An exponentially distributed service time with rate μ can be generated according to $s = -\frac{1}{\mu} \ln \zeta$, where ζ is a uniformly distributed random number in [0, 1]. Now let the service rate of a server, say server k, change from μ_k to $\mu_k + \Delta\mu_k$, $\Delta\mu_k << \mu_k$. Then the service time s will change to $s' = -\frac{1}{\mu_k + \Delta\mu_k} \ln \zeta$ (the same ζ is used for both s and s'). Thus, the perturbation of the service time induced by the change in μ_k is

$$\Delta s = s' - s \approx -\frac{\Delta\mu_k}{\mu_k} s. \qquad (1)$$

In summary, because of a small (infinitesimal) change of service rate $\Delta\mu_k$, every customer's service completion time at server k will obtain (be delayed by) a perturbation that is proportional to its original service time with a multiplier $-\frac{\Delta\mu_k}{\mu_k}$. This is the rule of *perturbation generation*.

Next, we observe that a perturbation will affect the service starting and completion times of other customers in the same server and in other servers in the network. We say a perturbation will be *propagated* through the network.

First, the perturbations generated at a server will accumulate until this server is idle. When the server enters an idle period, all its perturbations will be lost. (After the idle period, the service starting time is determined by the customer that terminates the idle period, which carries the perturbation of another server.) Second, when a customer finishes its service at server i and enters server j after an idle period of server j, server j will obtain the same amount of perturbations as server i. If server j is not idle at this time, the perturbation at server i will only affect the arrival time of this customer and will not affect any other customers/servers.

Figure 2 illustrates the perturbation generation and propagation process of a network in which the service rate of server 1 is decreased with an infinitesimal amount. Given a system sample path obtained by simulation or observation, we can record the perturbations of all servers as they are generated and propagated along the sample path. From the perturbations we can get a perturbed sample path and finally get the performance changes caused by the change in the service rate. Specifically, we have the following algorithm for the perturbations of the service completion times (if server k's service rate is perturbed).

In Step 2, we add $s_{k,l}$ as the perturbation generated, instead $-\frac{\Delta\mu_k}{\mu_k} s_{k,l}$ as indicated by (1).

Algorithm 1 Perturbation Analysis

1. *Initialization:* Set variables $\Delta_i = 0, i = 1, 2, \cdots, M$.
2. *Perturbation generation:* At the lth service completion time of server k, set $\Delta_k = \Delta_k + s_{k,l}, l = 1, 2, \cdots$, where $s_{k,l}$ is the service time of the lth customer at server k.
3. *Perturbation propagation:* If a customer from server i terminates an idle period of server j, set $\Delta_j = \Delta_i$, $i, j = 1, 2, \cdots, M$.

The factor $-\frac{\Delta \mu_k}{\mu_k}$ will be canceled when estimating performance derivatives with respect to $\Delta \mu_k$. The algorithm yields a perturbed sample path. With the original path and the perturbed path, we can estimate the original and perturbed (steady-state) throughput, then the estimates of its derivative with respect to μ_k can be obtained, and it is proved that the estimate is strongly consistent. Only a few lines need to be added in the simulation code to obtain the derivatives. Experiments show that the results are very accurate (error is around 5 %, compared with analytical results, Ho and Cao 1983).

Markov Systems

The decomposition principle shown in Fig. 1 has been applied to Markov systems (Cao 2007).

Consider an irreducible and aperiodic Markov chain $\mathbf{X} = \{X_n : n \geq 0\}$ on a finite state space $\mathcal{S} = \{1, 2, \cdots, M\}$ with transition probability matrix $P = [p(j|i)] \in [0, 1]^{M \times M}$. Let $\pi = (\pi_1, \ldots, \pi_M)$ be the vector representing its steady-state probabilities and $f = (f_1, f_2, \cdots, f_M)^T$ be the reward (or cost) vector, where "T" represents transpose. We have $Pe = e$, where $e = (1, 1, \cdots, 1)^T$ is an M-dimensional vector whose all components equal 1, and we have $\pi = \pi P$. We consider the long-term average (steady-state) performance defined as

$$\eta = E_\pi(f) = \sum_{i=1}^{M} \pi_i f_i$$

$$= \pi f = \lim_{L \to \infty} \frac{F_L}{L}, \quad w.p.1, \qquad (2)$$

where

$$F_L = \sum_{l=0}^{L-1} f(X_l).$$

Let P' be another ergodic transition probability matrix on the same state space. Suppose P changes to $P(\delta) = P + \delta Q = \delta P' + (1 - \delta)P$, with $\delta > 0$, $Q = P' - P = [q(j|i)]$, and the reward function f keeps the same. We have $Qe = 0$. The performance measure will change to $\eta(\delta) = \eta + \Delta \eta(\delta)$. The derivative of η in the direction of Q is defined as $\frac{d\eta(\delta)}{d\delta} = \lim_{\delta \to 0} \frac{\Delta \eta(\delta)}{\delta}$.

In this discrete-state Markov system, a perturbation means that the system is perturbed from one state i to another state j. For example, consider the case where $q(i|k) = \frac{1}{2}, q(j|k) = \frac{1}{2}$, and $q(l|k) = 0$ for all $l \neq i, j$. Suppose that these probabilities change to $q(i|k) = \frac{1}{2} + \delta$, $q(j|k) = \frac{1}{2} - \delta$, and $q'(l|k) = 0$ for all $l \neq i, j$. Then it may happen that at some time in the original sample path the system transits from state k to state i, but in the perturbed path it transits from state k to state j instead. In this case, we say that the change in transition probabilities induces a perturbation from i to j at this time. To study the effect of such a perturbation, we consider two independent sample paths $\mathbf{X} = \{X_n; n \geq 0\}$ and $\mathbf{X}' = \{X_n'; n \geq 0\}$ with $X_0 = i$ and $X_0' = j$; both of them have the same transition matrix P. The average effect of a perturbation from i to j, $i, j = 1, \cdots, M$, on F_L can be measured by the *perturbation realization factor* defined as

$$d(i, j) = \lim_{L \to \infty} E\left[\sum_{l=0}^{L-1} (f(X_l') - f(X_l)) | X_0 \right.$$

$$\left. = i, X_0' = j \right]. \qquad (3)$$

The matrix $D \in \mathcal{R}^{M \times M}$, with $d(i, j)$ as its (i, j)th element, is called a *realization matrix*. We can prove that D satisfies the equation (Cao 2007)

$$D - PDP^T = F, \qquad (4)$$

where $F = fe^T - ef^T$. Because $d(i, j) = -d(j, i)$, for any i, j, we may define a vector $g = (g(1), \cdots, g(M))^T$ such that

$$d(i, j) = g(j) - g(i), \qquad (5)$$

and $D = ge^T - eg^T$. g is called a *performance potential*, which can be estimated with many sample path-based algorithms, and it satisfies the Poisson equation (Cao 2007)

$$(I - P + e\pi)g = f. \qquad (6)$$

Intuitively, (3) and (5) indicate that every visit to state i contributes to F_L on the average by the amount of $g(i)$, so the effect of a perturbation from i to j is $d(i, j) = g(j) - g(i)$. Now, we consider a sample path consisting of L transitions. Among these transitions, on the average there are $L\pi_i$ transitions at which the system is at state i. After being at state i, the system jumps to state j on the average $L\pi_i p(j|i)$ times. If the transition probability matrix P changes to $P(\delta) = P + \delta Q$, then the change in the number of visits to state j after being at state i is $L\pi_i q(j|i)\delta = L\pi_i[p'(j|i) - p(j||i)]\delta$. This contributes a change of $\{L\pi_i[p'(j|i) - p(j|i)]\delta\}g(i)$ to F_L. Thus, the total change in F_L due to the change of P to $P(\delta)$ is

$$\Delta F_L = \sum_{i,j=1}^{M} L\pi_i[p'(j|i) - p(j|i)]\delta g(i)$$
$$= L(\pi Q g)\delta.$$

Finally, we have

$$\frac{d\eta}{d\delta} = \lim_{\delta \to 0} \frac{1}{\delta}\frac{\Delta F_L}{L} = \pi Q g = \pi(P' - P)g. \qquad (7)$$

If the reward function also changes from f to $f(\delta) = f + \delta(f' - f)$, then (7) becomes

$$\frac{d\eta}{d\delta} = \pi[(P'g + f') - (Pg + f)]. \qquad (8)$$

Further Works

The ideas described in the previous subsections may stimulate new research topics in theoretical analysis, estimation algorithms, and applications. Here we can only give a very brief review for some of them.

1. There is a large literature on whether the PA-based derivative estimates are unbiased (finite period) and/or strongly consistent (steady state). This was first formulated in Cao (1985) and was further discussed in Heidelberger et al. (1988). By now, there have been extensive studies in this direction: proving the unbiasedness or consistency for various systems and modifying the approach for system when the IPA estimates are not unbiased (e.g., Cassandras and Lafortune 1999; Fu and Hu 1997; Glasserman 1991). This also includes the recently proposed fluid model; see ▸ Perturbation Analysis of Discrete Event Systems.

2. Another research topic is how to develop fast and efficient algorithms for estimating the performance gradients, especially in the case of Markov systems; see e.g., Cao and Wan (1998), Baxter and Bartlett (2001), and Cao (2005). This is called *policy gradients* in the reinforcement learning literature.

3. There are also research works on how the gradient estimates and policy iteration (see section "Direct Comparison and Policy Iteration") can be combined with stochastic approximation approaches to develop fast convergent optimization algorithms; see Marbach and Tsitsiklis (2001) for gradient-based approach and Fang and Cao (2004) for policy iteration-based approach.

Direct Comparison and Policy Iteration

The sensitivity-based view has been extended to optimization in discrete spaces of policies. With this view, we can develop a new approach to performance optimization based on a direct comparison of the performance of any two policies. This provides an alternative to the standard dynamic programming to solving the Markov decision processes (MDP) types of problems; it also has been applied to solve some problems when the standard MDP fails (Cao 2007; Cao and Wan 2013).

P

In an MDP, there is an action space denoted as $\mathcal{A}$. For simplicity, we only consider the discrete case. When the system is at any state $i \in \mathcal{S}$, an action $\alpha = d(i)$ is taken, which controls the system transition probability, denoted as $p^\alpha(j|i)$, $i, j \in \mathcal{S}$, and the reward function, denoted as $f(i, \alpha)$. The mapping $d : \mathcal{S} \to \mathcal{A}$ is called a *policy*. Since a policy corresponds to a transition probability matrix $P^d = [p^{d(i)}(j|i)]_{i,j=1}^M$ and the reward vector $f^d = (f(1, d(1)), \cdots, f(M, d(M)))^T$, we also call the pair (P, f) a policy.

Consider two policies (P, f) and (P', f'), and assume that the Markov chain under both policies is ergodic. We use prime " $'$ " to denote the quantities associated with (P', f'). First, multiplying both sides of the Poisson equation (6) with π' on the left and after some calculations, we get

$$\eta' - \eta = \pi'\{(P'g + f') - (Pg + f)\}. \quad (9)$$

This is called a *performance difference formula*, and many optimization results can be derived from it in an intuitive way.

Policy Iteration and the Optimality Equation

The difference formula (9) has a nice decomposition structure: it contains two factors, the first one π', which does not depend on P, and the second one $(P'g + f') - (Pg + f)$, in which all the parameters are known except the performance potential g, which can be obtained by only analyzing system with P. This nice feature makes the difference formula the basis of performance optimization.

For two M-dimensional vectors a and b, we define $a = b$ if $a(i) = b(i)$ for all $i = 1, 2 \cdots, M$; $a \leq b$ if $a(i) \leq b(i)$ for all $i = 1, 2 \cdots, M$; $a < b$ if $a(i) < b(i)$ for all $i = 1, 2 \cdots, M$; and $a \preceq b$ if $a(i) < b(i)$ for at least one i and $a(j) = b(j)$ for other components. The relation $\leq$ includes $=$, $\preceq$, and $<$. Similar definitions are used for the relations $>$, $\succeq$, and $\geq$.

Next, we note that $\pi'(i) > 0$ for all $i = 1, 2, \cdots, M$. Thus, from (9), we know that if $(P' - P)g + (f' - f) \succeq 0$, then $\eta' - \eta > 0$.

From (9) and the fact $\pi' > 0$, the proof of the following lemma is straightforward.

Lemma 1 *If $Pg + f \preceq (\leq) P'g + f'$, then $\eta < (\leq) \eta'$.*

It is interesting to note that in the lemma, we use only the potentials with one Markov chain, i.e., g. Thus, because of the special structure of the performance difference formula (9), if the condition in Lemma 1 holds, to compare the performance measures under two policies, we may only need the potentials with one policy.

Policy iteration and the optimality equation can be easily derived from (9) and Lemma 1.

Algorithm 2 Policy Iteration

1. Guess an initial policy d_0, and set $k = 0$.
2. (Policy evaluation) Obtain the potential g^{d_k} by solving the Poisson equation $(I - P^{d_k})g^{d_k} + \eta^{d_k}e = f^{d_k}$ or estimating it on a sample path. (The superscript "d_k" is added to quantities associated with policy d_k.)
3. (Policy improvement) Choose

$$d_{k+1} \in \arg\left\{\max_{d \in \mathcal{D}}\left[f^d + P^d g^{d_k}\right]\right\}, \quad (10)$$

component-wise (i.e., to determine an action for each state). If in state i, action $d_k(i)$ attains the maximum, and set $d_{k+1}(i) = d_k(i)$.
4. If $d_{k+1} = d_k$, stop; otherwise, set $k := k + 1$ and go to Step 2.

It follows directly from Lemma 1 that when the iteration does not stop, the performance improves at each iteration. It can be proved easily by construction that the iteration stops at an optimal policy. Again, from Lemma 1, when the iteration stops at a polict $\hat{d}$, it holds

$$\eta^{\hat{d}}e + g^{\hat{d}} = \max_{d \in \mathcal{D}}\left\{f^d + P^d g^{\hat{d}}\right\}. \quad (11)$$

This is the Hamilton-Jacobi-Bellman (HJB) optimality equation. If we further define the Q-factor,

$$Q^d(i, \alpha) = f(i, \alpha) + \sum_j p^\alpha(j|i)g^d(j)]. \quad (12)$$

Then the policy iteration equation (10) and the HJB equation (11) become

$$d_{k+1}(i) := \arg \max_{\alpha \in \mathcal{A}} \{Q^{d_k}(i, \alpha)\} \qquad (13)$$

and

$$Q^{\hat{d}}(i, \alpha) = \max_{\beta \in \mathcal{A}} \{Q^{\hat{d}}(i, \beta)\}. \qquad (14)$$

The only difference between (8) and (9) is that π in (8) is replaced by π' in (9). This leads to an interesting observation: *policy iteration in MDPs in fact chooses the policy with the steepest directional derivative as the policy in the next iteration.* Therefore, policy iteration in fact can be viewed as the "gradient-based" optimization in a discrete space.

In our approach, the HJB equation and policy iteration are obtained from the performance difference equation (9), which compares the performance of any two policies. It is hence called a *direct-comparison* based approach. This approach also applies to more general problems, such as multichain Markov systems, systems with absorbing states, and problems with other performance criteria such as the discounted performance and the bias, etc.

The direct-comparison approach has been successfully applied to the *n*th-bias optimality problem (Cao 2007). Essentially, starting with the performance difference formulas (those similar to (9) for different performance), we can develop a simple and direct approach to derive the results that are equivalent to the sensitive discount optimality for multichain Markov systems with long-run average criteria (Puterman 1994), and no discounting is needed and no dynamic programming is used. The approach, motivated by the development for discrete event dynamic systems, provides a clear overall picture for the area of MDP.

The direct-comparison approach can also be applied to some problems where dynamic programming fails, including the event-based optimization problems, where the sequence of events may not be Markovian; see the next section.

Event-Based Optimization

It is well known that for most systems modeled by Markov processes, the state spaces are too large, and it is not computationally feasible to implement policy iteration or to solve the HJB equations. On the other hand, in many practical problems in engineering, finance, and social sciences, control actions are only taken when certain events occur. For example, in the traffic control of a network of subnetworks, often times one cannot control the traffic in the same subnetwork, and control actions are only applied when there are packets transferring among different subnetworks. In a portfolio management problem, the investor sells or buys stocks when the price history experiences some predetermined patterns (e.g., reaches some level). In a sensor network, actions are taken only when one of the sensors detects some abnormal situations. In a material handling problem, actions are taken when the inventory level falls below certain threshold.

Conceptually, anything happened in the past can be chosen as an event. However, the number of such events is too big (much bigger than the number of states), and studying all these events makes analysis infeasible and defeats our original purpose. Therefore, to properly define events, we need to strike a balance between the generality on one hand and the applicability on the other hand.

In the event-based setting, *an event e is defined as a set of state transitions with certain common properties.* That is, $e := \{\langle i, j \rangle : i, j \in \mathcal{S}$ and $\langle i, j \rangle$ has common properties}, where $\langle i, j \rangle$ denotes a state transition from i to j. This definition can also be easily generalized to represent a finite sequence of state transitions. We shall see that in many real problems, the number of events requiring control actions is usually much smaller than that of the states.

An event-based policy d is defined as a mapping from $\mathcal{E}$ to $\mathcal{A}$, with $\mathcal{E}$ being the space of all events. That is, $d : \mathcal{E} \rightarrow \mathcal{A}$. Let $\mathcal{D}_e$ denote the set of all the stationary and deterministic policies. The reward function under policy d is denoted as $f^d = f(i, d(e))$, and the associated long-run average performance is denoted as η^d. When an event e happens, we choose an action $a = d(e)$ according to a policy d, where $e \in \mathcal{E}$ and $d \in \mathcal{D}_e$. Our goal is to find an optimal policy $\hat{d}$ which maximizes the long-run average performance as follows.

$$\hat{d} = \arg\max_{d \in \mathcal{D}_e} \{\eta^d\} \qquad (15)$$

The main difficulty in developing the event-based optimization theories and algorithms lies in the fact that the sequence of events is usually not Markovian. The standard optimization approach such as dynamic programming does not apply to such problems. However, we can apply the direct-comparison approach to this event-based optimization problem. Here we give a very brief discussion.

Consider an event-based policy d, $d \in \mathcal{D}_e$. When event e occurs, the conditional transition probability is denoted as $p^{d(e)}(j|i,e)$. Let $\pi^d(i|e)$ be the conditional steady-state probability of state i when event e occurs under policy d. Define the aggregated Q-factor

$$Q^d(e, \alpha) = \sum_i \pi^d(i|e)$$
$$\times \left[f(i, \alpha) + \sum_j p^\alpha(j|i,e)g^d(j) \right]. \quad (16)$$

We may use these aggregated Q-factors to develop an event-based policy iteration algorithms as Algorithm 2 and obtain the policy iteration equation (cf. (13))

$$d_{k+1}(e) := \arg\max_{\alpha \in \mathcal{A}} \{Q^{d_k}(e, \alpha)\}, \qquad e \in \mathcal{E}, \quad (17)$$

and the event-based HJB optimality equation (cf. (14)) is

$$Q^{\hat{d}}(e, \alpha) = \max_{\beta \in \mathcal{A}} \{Q^{\hat{d}}(e, \beta)\}. \qquad (18)$$

It can be proved that if the conditional probability $\pi^h(i|e)$ does not depend on the policy; i.e.,

$$\pi^h(i|e) = \pi^d(i|e), \quad \forall i, e, \quad \text{and } h, d, \quad (19)$$

then policy iteration (17) indeed leads to a sequence of increasing performance and (18) specifies an event-based optimal policy.

The event-based aggregated Q-factor (16) can be estimated on a sample path in the same way as for potentials (Cao 2007). The number of events requiring actions is usually much smaller than the

number of states, and in some cases such as the network admission control problem, it is linear to the system size.

The crucial condition for the above event-based optimization is (19). There are many problems, such as the control of the networks of networks and the portfolio management problem, for which the condition holds; there are also many problems for which the condition does not hold. The error in applying (18) and policy iteration (17) comes from the difference between $\pi^h(i|e)$ and $\pi^d(i|e)$ at each iteration. Further research is needed.

We use a computer network as an example, in which each computer or router can be modeled as a queueing subnetwork and the computer network is then a network of such subnetworks. Assume that there is an M subnetwork, and subnetwork m, $m = 1, 2, \cdots, M$, consists of k_m servers. The number of customers at the jth server of subnetwork m is denoted as $n_{m,j}$, $j = 1, 2, \cdots, k_m$, and the number of customers in all the servers in subnetwork m is denoted as $N_m = \sum_{j=1}^{k_m} n_{m,j}$. Suppose the service time is exponentially distributed, then the system state is $\mathbf{n} := (n_{1,1}, \cdots, n_{1,k_1}; \cdots; n_{M,1}, \cdots, n_{M,k_M})$, and the aggregated state is $\mathbf{N} := (N_1, \cdots, N_M)$. Suppose that the transition probabilities among the servers in the same subnetwork are fixed and we can only control the transition probabilities among the subnetworks, and furthermore, we can only observe $\mathbf{N}$. Then the problem can be modeled as an event-based optimization with an event being a customer transition among subnetworks, and meanwhile, the aggregated state is $\mathbf{N}$. It can be proved that the condition (19) holds, and therefore, we may apply policy iteration (17) and HJB equation (18).

Many existing problems fit the event-based framework. For example, in a partially observable Markov decision process (POMDP), we may define an observation, or a sequence of observations, as an event. Other examples include state and time aggregations, hierarchical control (hybrid systems), and options. Different events can be defined to capture the special features in these different problems. In this sense, the event-based

approach may provide a unified view to these different problems (Cao 2007).

The Sensitivity-Based Approach and Optimal Control

We refer to the approaches discussed in the previous sections the *sensitivity-based approach*. When the parameters are continuous, it is the gradient-based approach, and the focus is on developing efficient algorithms that utilize the particular system structure to estimate the gradients (to justify the algorithms, the unbiasedness or the consistency of the estimates should be proved). When the parameters are discrete, it is the direct-comparison approach that leads to policy iteration and the HJB equations, and policy iteration can be viewed as the gradient-based method in discrete spaces. This approach is different from the conventional dynamic programming, and it has been successfully applied to MDP with different criteria and the *n*-bias optimization problems as well as the event-based optimization and some other problems that dynamic programming fails.

This sensitivity-based approach was motivated by the study of discrete event dynamic systems (DEDS). It has been realized that the principles and methodologies developed for DEDSs also apply to the optimization of continuous-time and continuous-state (CTCS) systems. Here are some examples:

1. *CTCS systems:* For CTCS systems, the dynamic is driven by Brownian motions or Levy processes. The transition probability matrix in DEDS should be replaced by the infinitesimal generator in CTCS, which is an operator on the space of continuous functions. With the performance difference formulas, we can redevelop the stochastic optimal control theory for many performance measures, including the long-run average, discounted performance, and finite horizon problems, with no dynamic programming; see Cao et al. (2011).

2. *Time-inconsistent optimization problems:* In behavioral finance, people's preference is modeled with a distorted probability. For example, a risk-taking person buys lotteries, because in her/his mind, s/he enlarges the possibility of winning a large sum, and a risk averse person buys insurance, because s/he is afraid of a big loss and therefore enlarges its probability.

The optimization problem with a distorted probability suffers from the time-inconsistent issue; i.e., an optimal policy for the problem in period $[t, T]$, $0 < t < T$, is not optimal in the same period for the problem in $[0, T]$. Thus, the standard dynamic programming fails.

The gradient-based approach has been applied to the portfolio management problem with probability distortion. With this approach, we discovered that the performance with distorted probability maintains some sort of linearity called *mono-linearity*. This property shed new insights to the portfolio management problem and the nonlinear expected utility theory (Cao and Wan 2013).

Conclusion

A sensitivity-based approach has been developed to the performance optimization of discrete event dynamic systems (DEDS). The approach utilizes the dynamic structure of DEDS. For systems with continuous parameters, it is the gradient-based optimization, in which the special feature of a DEDS helps in developing efficient algorithms to estimate the performance derivatives; for systems with discrete policies, it is the direct-comparison-based approach, with which policy iteration and HJB equations can be derived intuitively by using the performance difference formulas. Policy iteration can be viewed as the gradient method in a discrete space. The estimation of gradients and the implementation of policy iteration can be carried out on a given sample path, and efficient online learning algorithms can be developed (Cao 2007).

The sensitivity-based approach was developed for DEDSs, but its principle also applies to systems with continuous-state spaces. The approach provides an alternative to the traditional dynamic programming and therefore can be applied to some problems where dynamic programming does not work.

Cross-References

▶ Models for Discrete Event Systems: An Overview
▶ Perturbation Analysis of Discrete Event Systems

Bibliography

Baxter J, Bartlett PL (2001) Infinite-horizon policy-gradient estimation. J Artif Intell Res 15: 319–350

Cao XR (1985) Convergence of parameter sensitivity estimates in a stochastic experiment. IEEE Trans Autom Control 30:834–843

Cao XR (2005) A basic formula for online policy gradient algorithms. IEEE Trans Autom Control 50(5):696–699

Cao XR (2007) Stochastic learning and optimization – a sensitivity-based approach. Springer, New York

Cao XR, Wan YW (1998) Algorithms for sensitivity analysis of Markov systems through potentials and perturbation realization. IEEE Trans Control Syst Technol 6:482–494

Cao XR, Wan XW (2013) Analysis of non-linear behavior – a sensitivity-based approach. submitted

Cao XR, Wang DX, Lu T, Xu YF (2011) Stochastic control via direct comparison. Discret Event Dyn Syst Theory Appl 21:11–38

Cassandras CG, Lafortune S (1999) Introduction to discrete event systems. Kluwer Academic Publishers, Boston

Fang HT, Cao XR (2004) Potential-based on-line policy iteration algorithms for Markov decision processes. IEEE Trans Autom Control 49:493–505

Fu MC, Hu JQ (1997) Conditional Monte Carlo: gradient estimation and optimization applications. Kluwer Academic Publishers, Boston

Glasserman P (1991) Gradient estimation via perturbation analysis. Kluwer Academic Publishers, Boston

Heidelberger P, Cao XR, Zazanis M, Suri R (1988) Convergence properties of infinitesimal perturbation analysis estimates. Manag Sci 34:1281-1302

Ho YC, Cao XR (1983) Perturbation analysis and optimization of queueing networks. J Optim Theory Appl 40:559–582

Ho YC, Cao XR (1991) Perturbation analysis of discrete-event dynamic systems. Kluwer Academic Publisher, Boston

Marbach P, Tsitsiklis TN (2001) Simulation-based optimization of Markov reward processes. IEEE Trans Autom Control 46:191–209

Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, New York

Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407

PID Control

Sebastian Dormido[1] and Antonio Visioli[2]
[1]Departamento de Informatica y Automatica, UNED, Madrid, Spain
[2]Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia, Italy

Synonyms

Proportional-Integral-Derivative Control

Abstract

Since their introduction in industry a century ago, proportional–integral–derivative (PID) controllers have become the de facto standard for the process industry. In this entry, fundamentals of PID control are outlined, starting from the basic control law. Additional functionalities and the tuning and automatic tuning of the parameters are then considered.

Keywords

Anti-windup; Autotuning; Controller tuning; Derivative action; PI control; Proportional control; Proportional-integral-derivative control; Ziegler-Nichols

Introduction

A proportional–integral–derivative (PID) controller is a three-term controller that has a long history in the automatic control field, starting from the beginning of the last century. Owing to its intuitiveness and relative simplicity, in addition to the satisfactory performance that it is

able to provide with a wide range of processes, it has become the de facto standard controller in industry. It has been evolving along with the progress of technology, and nowadays it is very often implemented in digital form rather than with pneumatic or electrical components. It can be found in virtually all kinds of control equipments, either as a stand-alone (single-station) controller or as a functional block in Programmable Logic Controllers (PLCs) and Distributed Control Systems (DCSs). Actually, the new potentialities offered by the development of the digital technology and of the software packages have led to a significant growth of the research in the PID control field: new effective tools have been devised for the improvement of the analysis and design methods of the basic algorithm as well as for the improvement of the additional functionalities that are implemented with the basic algorithm in order to increase its performance and its ease of use.

The success of the PID controllers is also enhanced by the fact that they often represent the fundamental component for more sophisticated control schemes that can be implemented when the basic control law is not sufficient to achieve the required performance or when a more complicated control task is of concern.

Basics

Using a PID controller means applying a feedback controller that consists of the sum of three types of control actions: a proportional action, an integral action, and a derivative action.

The proportional control action is proportional to the current control error, according to the expression

$$u(t) = K_p e(t) = K_p(r(t) - y(t)), \quad (1)$$

where u is the controller output, K_p is the proportional gain, r is the reference signal, and y is the process output. Its meaning is straightforward, since it implements the typical operation of increasing the control variable when the control error is large (with appropriate sign). The transfer

function of a proportional controller can be trivially derived as

$$C(s) = K_p. \quad (2)$$

The main drawback of using a pure proportional controller is that, in general, it cannot set to zero the steady-state error. This motivates the addition of a bias (or reset) term u_b, namely,

$$u(t) = K_p e(t) + u_b. \quad (3)$$

The value of u_b can then be adjusted manually until the steady-state error is reduced to zero.

In commercial products, the proportional gain is often replaced by the proportional band PB, which is the range of error that causes a full-range change of the control variable, i.e.,

$$PB = \frac{100}{K_p}. \quad (4)$$

The integral action is proportional to the integral of the control error, i.e.,

$$u(t) = K_i \int_0^t e(\tau) d\tau, \quad (5)$$

where K_i is the integral gain. It appears that the integral action is related to the past values of the control error. The corresponding transfer function is

$$C(s) = \frac{K_i}{s}. \quad (6)$$

The presence of an integral action allows to reduce the steady-state error to zero when a step reference signal is applied or a step load disturbance occurs. In other words, the integral action is able to set automatically the correct value of u_b in (3) so that the steady-state error is zero. For this reason, the integral action is also often called *automatic reset*.

While the proportional action is based on the current value of the control error and the integral action is based on the past values of the control error, the derivative action is based on the predicted future values of the control error. An ideal derivative control law can be expressed as

P

$$u(t) = K_d \frac{de(t)}{dt}, \qquad (7)$$

where K_d is the derivative gain. The corresponding controller transfer function is

$$C(s) = K_d s. \qquad (8)$$

The meaning of the derivative action can be better understood by considering the first two terms of the Taylor series expansion of the control error at time T_d ahead:

$$e(t + T_d) \simeq e(t) + T_d \frac{de(t)}{dt}. \qquad (9)$$

If a control law proportional to this expression is considered, i.e.,

$$u(t) = K_p \left(e(t) + T_d \frac{de(t)}{dt} \right), \qquad (10)$$

this naturally results in a PD controller. The control variable at time t is therefore based on the predicted value of the control error at time $t + T_d$. For this reason, the derivative action is also called *anticipatory control*, or *rate action*, or *pre-act*.

The combination of the proportional, integral, and derivative actions can be done in different ways. In the so-called *ideal* or *non-interacting* form, the PID controller is described by the following transfer function:

$$C_i(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right), \qquad (11)$$

where K_p is the proportional gain, T_i is the integral time constant, and T_d is the derivative time constant. An alternative representation is the *series* or *interacting form*:

$$C_s(s) = K_p' \left(1 + \frac{1}{T_i' s} \right) \left(T_d' s + 1 \right)$$

$$= K_p' \left(\frac{T_i' s + 1}{T_i' s} \right) \left(T_d' s + 1 \right), \quad (12)$$

where the fact that a modification of the value of the derivative time constant T_d' affects also the proportional action justifies the nomenclature

adopted. Suitable conversion formulae can be applied to obtain an ideal PID controller equivalent to a series one. Obtaining an equivalent PID controller in series form starting from an ideal one is possible only if the zeros of the ideal PID controller are real.

Additional Functionalities

The expression (11) or (12) of a PID controller is actually not employed in practical cases because of a few problems that can be solved with suitable modifications of the basic control law.

Modifications of the Derivative Action
From Expressions (11) and (12), it appears that the controller transfer function is not proper, because of the derivative action, and therefore, it cannot be implemented in practice. Indeed, the high-frequency gain of the pure derivative action is responsible for the amplification of the measurement noise in the manipulated variable. This problem can be solved by filtering the derivative action with (at least) a first-order low-pass filter. The filter time constant should be selected in order to suitably filter the noise and to avoid a significant influence on the dominant dynamics of the PID controller. Thus, it can be selected as T_d/N, where N generally assumes a value between 1 and 33, although in the majority of the practical cases its setting falls between 8 and 16. Alternatively, the overall control variable can be filtered.

Another issue related to the derivative action that has to be considered is the so-called derivative kick. In fact, when an abrupt (stepwise) change of the set-point signal occurs, the derivative action is very large, and this results in a spike in the control variable signal, which is undesirable. This problem could be simply avoided by applying the derivative term to the process output only instead of the control error. In this case, the ideal (not filtered) derivative action becomes

$$u(t) = -K_p T_d \frac{dy(t)}{dt}. \qquad (13)$$

Obviously, when the set-point signal is constant, applying the derivative term to the control error or to the process variable is equivalent. Thus, the load disturbance rejection performance is the same in both cases.

Set-Point Weighting for Proportional Action

A typical problem with the design of a feedback controller is to achieve a high performance both in the set-point following task and in the load disturbance rejection task at the same time. For example, for stable processes, a fast load disturbance rejection is achieved with a high-gain (aggressive) controller, which gives an oscillatory set-point step response on the other side. This problem can be approached by using a two-degree-of-freedom control architecture, where a feedback controller is designed to achieve a high bandwidth and therefore a satisfactory load disturbance rejection performance, and then the set-point signal is filtered before applying it to the closed-loop system.

In the context of PID control, this can be achieved by weighting the set-point signal for the proportional action, that is, to define the proportional action as follows:

$$u(t) = K_p(\beta r(t) - y(t)), \qquad (14)$$

where the value of β is between 0 and 1.

In this way, the control scheme has a feedback controller (11), and the set-point signal is filtered by the system

$$F(s) = \frac{1 + \beta T_i s + T_i T_d s^2}{1 + T_i s + T_i T_d s^2}. \qquad (15)$$

The load disturbance rejection task is decoupled from the set-point following task, and obviously it does not depend on the weight β, which can be employed to smooth the (step) set-point signal in order to damp the response to a set-point change. The smaller the value of β, the smaller the overshoot and the higher the rise time.

Anti-windup

One of the most well-known possible sources of performance degradation is the so-called integrator windup phenomenon, which occurs when the controller output saturates (typically when a large set-point change occurs). In this case, the system operates as in the open-loop case, since the actuator is at its maximum (or minimum) limit, regardless of the process output value. The control error decreases more slowly than in the ideal case (where there are no saturation limits), and therefore, the integral term becomes large (it *winds up*). Thus, even when the value of the process variable attains that of the reference signal, the controller still saturates due to the integral term, and this generally yields large overshoots and settling times.

In order to cope with this problem, an additional functionality designed for this purpose can be conveniently used. This can be done in different ways. For example, in the conditional integration approach, the integration is stopped when the control variable saturates and the control error and the control variable have the same sign. Alternatively, in the back-calculation approach, the integral term is recomputed when the controller saturates by feeding back the difference of the saturated and unsaturated control signal.

Tuning

The selection of the PID parameters, i.e., the tuning of the PID controller, is obviously the crucial issue in the overall controller design. This operation should be performed in accordance with the control specifications (which should take into account the set-point following, the load disturbance rejection, the control effort, and the robustness of the system). A major advantage of the PID controller is that its parameters have a clear physical meaning, and therefore, manual tuning is relatively simple. For example, for stable processes, increasing the proportional gain leads, in general, to a faster but more oscillatory response. In fact, by increasing K_p for the same value of the control error, the proportional control action increases, and so does the aggressiveness

of the controller. Then, increasing the integral time constant (i.e., decreasing the effect of the integral action) results, in general, in a slower response but in a more damped system. This is because a larger value of T_i implies a smaller value of the control action at a given time instant of the transient response (assuming the same values of the past control errors). Finally, increasing the derivative time constant gives a damping effect. Indeed, if the set point is constant, the derivative action is proportional to the derivative of the process variable with a negative sign, and therefore, the derivative action increases (with a negative sign) when the slope of the transient response increases (so that a big overshoot is avoided). However, in this context, much care should be taken to avoid increasing the derivative time constant too much as an opposite effect might occur in this case and an unstable system could eventually result. This is because the prediction over a too long time interval might be wrong.

The above considerations can be understood by considering a transient response in the time domain but also by considering the frequency response of the system and how it changes by modifying the PID parameters. For example, the effect of increasing the three controller actions can be seen as translating any point of the Nyquist plot in each of the directions shown in Fig. 1.

From another point of view, analogous considerations can be done by considering the Bode plot. For example, considering a PID controller in ideal form with a filter on the overall control action, the effect of modifying the three parameters in the controller Bode plot is shown in Fig. 2. The effects of the parameter modification in the achieved performance can be better ascertained by plotting the frequency response of the loop transfer function for different cases. As an example, consider the process $P(s) = 1/(10s + 1)e^{-4s}$ and the PID controller $C(s) = K_p(1 + \frac{1}{T_i s} + T_d s)\frac{1}{T_f s + 1}$ where the parameters are in the following ranges: $K_p \in [2, 15/4]$, $T_i \in [4, 16]$, $T_d \in [1.5, 4]$, being always $T_f = 0.1$. From Fig. 3, it appears that an increment of K_p yields an increment of the bandwidth and a decrement of the phase margin. Conversely, an increment of T_i has an opposite effect. Finally, it can be seen that the increment of T_d initially yields to an increment of the bandwidth and of the phase margin at the same time, but then a sudden increment of the bandwidth occurs, and this corresponds to a sudden decrement of the phase margin (because of the dead time of the process) with a possible loss of stability.

In any case, in order to ease the procedure, a large number of tuning rules have been proposed in the last century, starting from the well-known Ziegler–Nichols ones (Nichols and Ziegler 1942).

PID Control, Fig. 1 Effect of an increment of the three PID control actions on a point of the Nyquist plot
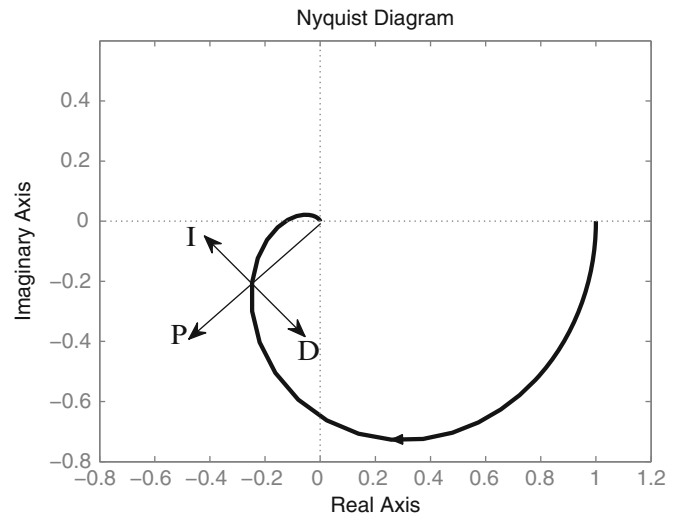
Different approaches in this context are analyzed hereafter.

Empirical Tuning

Empirical tuning methods (like the Ziegler–Nichols and its refinements, Cohen–Coon, or Chien–Hrones–Reswick ones (O'Dwyer 2006)) consist in selecting the parameters of the PID controllers by using some empirical formulae which give the PID gains based on parameters of the process. Usually, the process parameters are those of a first-order-plus-dead-time (FOPDT) model or the ultimate gain and frequency of the process itself (the ultimate gain is the largest value of a proportional-only control that produces a sustained oscillation of the process variable, that is, that results in a marginally stable closed-loop system, while the ultimate frequency is the frequency of the corresponding sustained oscillation). These parameters can be obtained by means of a simple open-loop (step response) or closed-loop (relay feedback) experiment.

Model-Based Tuning

In model-based tuning (like the Dahlin's and Haalman's methods and the Internal Model Control one (O'Dwyer 2006)), the PID control law is determined analytically starting from a process model and by selecting an appropriate (closed-loop) target transfer function. The user generally selects the desired closed-loop time constant as a tuning parameter which allows the handling of the trade-off between aggressiveness and robustness (and control effort). In this context, according to the well-known SIMC (Simplified Internal Model Control) tuning rules (Skogestad 2003), the closed-loop time constant should be selected equal to the dead time of the process.

Optimal Tuning

Optimal tuning rules aim at minimizing a given objective function. Usually, an integral function of the control error is selected for this purpose, for example,

$$J = \int_0^\infty t^n e^2(t) dt \qquad (16)$$

where $n = 0, 1, 2$ or the Integrated Absolute Error

$$IAE = \int_0^\infty |e(t)| dt. \qquad (17)$$

By solving the optimization problem for different kinds of (normalized) processes and by interpolating the results, it has been possible to obtain tuning formulae that give the PID gains based on the process parameters. Actually, it should be noted that as no (robustness) constraints are considered in the optimization procedure, a poor robustness may eventually result in the control system.

Robust Tuning

Recently, tuning rules which explicitly consider the robustness issue have been devised. In particular, the maximum of the sensitivity function is often considered as robustness index. A selected value of M_s is then employed as a constraint in finding the optimal PID parameters which minimize a given performance index. An additional constraint on the maximum complementary sensitivity function can also be considered. For example, the AMIGO tuning rules (Åström and Hägglund 2004) have been devised by applying this approach, where the integral gain is maximized in order to obtain the best reduction of load disturbances.
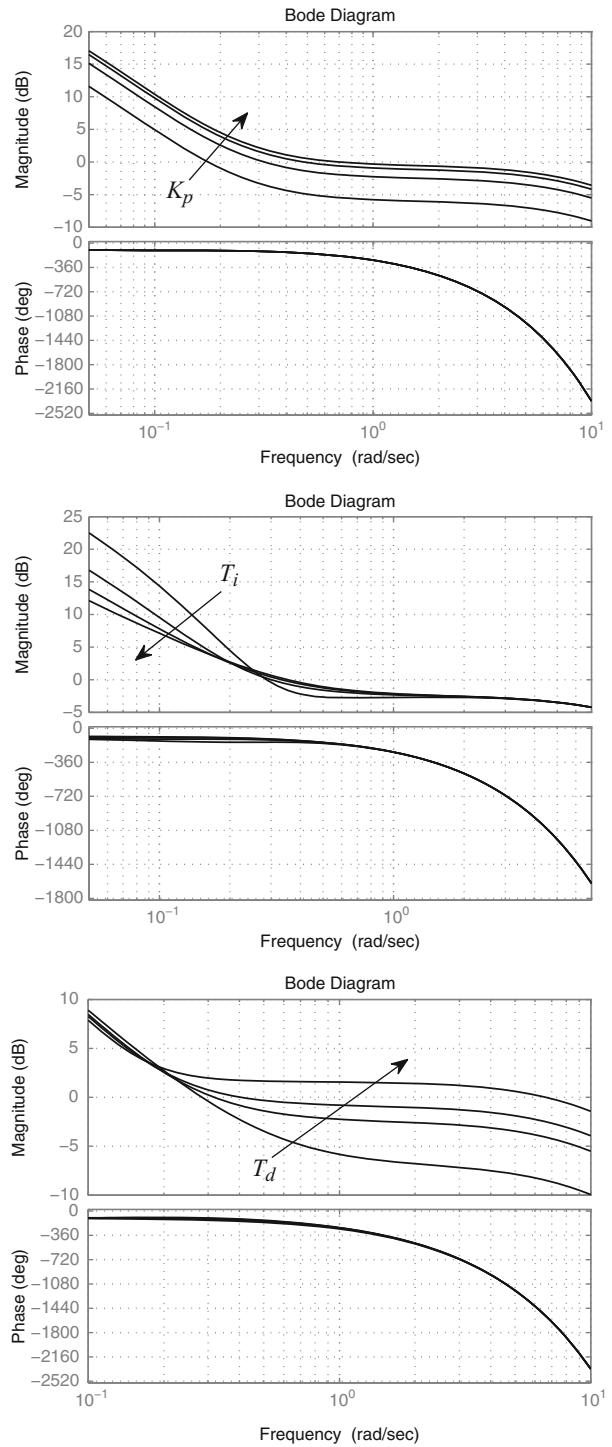
Automatic Tuning

The functionality of automatically identifying the process model and tuning the controller based on that model is called automatic tuning (or, simply, auto-tuning) (▶ Autotuning). In particular, an identification experiment is performed after an explicit request of the operator, and the values of the PID parameters are updated at the end of it (for this reason, the overall procedure is also called *one-shot automatic tuning* or *tuning-on-demand*). The design of an automatic tuning procedure involves many critical issues, such as the choice of the identification procedure (usually based on an open-loop step response or on a relay feedback experiment), of the a priori selected

PID Control, Fig. 2 Effect
of an increment of three
PID parameters on the
controller Bode plot. The
modifications are
considered separately,
namely, two parameters are
fixed, while the other is
modified
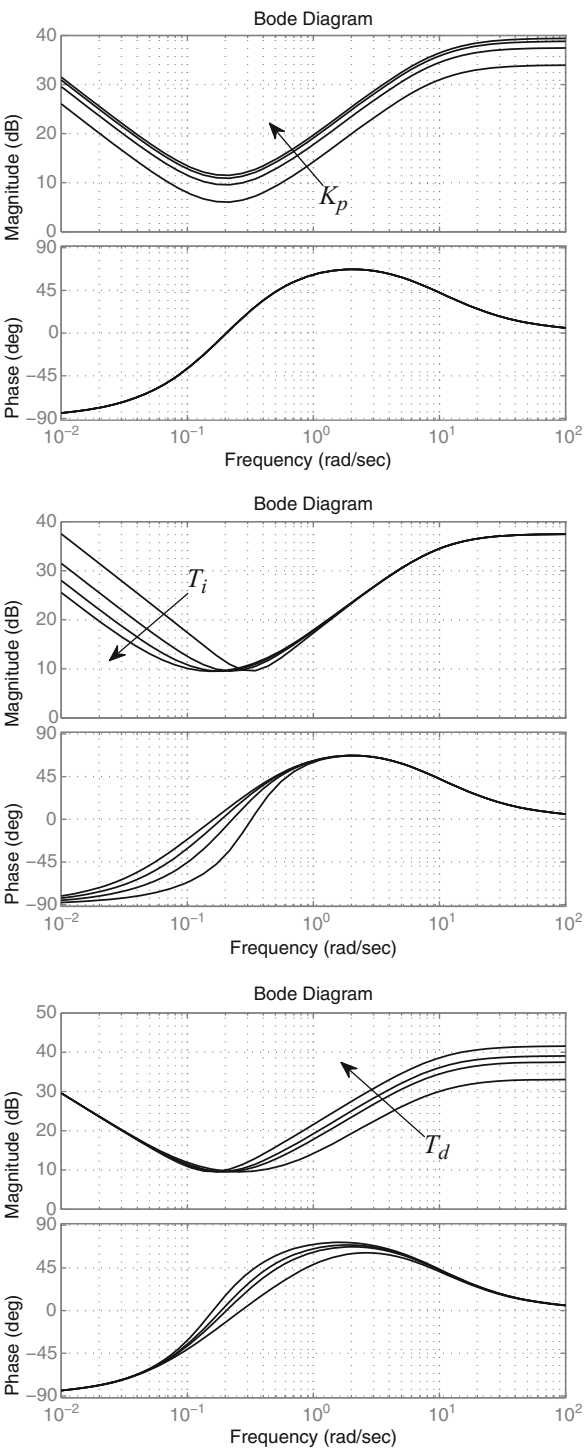
PID Control, Fig. 3
Example of the effect of an
increment of three PID
parameters on the loop
transfer function Bode plot.
The modifications are
considered separately,
namely, two parameters are
fixed, while the other is
modified

(parametric or non parametric) process model, and of the tuning rule.

The one-shot automatic tuning functionality is available in practically all the single-station controllers available on the market. More advanced control units might provide a *self-tuning* functionality, where the identification procedure is continuously performed during routine process operation in order to track possible changes of the system dynamics and the PID parameters values are adaptively modified. In this case, all the issues related to adaptive control have to be taken into account. In particular, performance assessment methodologies, which are capable to evaluate if the PID design can be improved, are of significant relevance in this context (▶ Controller Performance Monitoring).

Design Tools

Although one of the major advantages of PID controllers is their relative simplicity, Computer-Aided Control System Design tools (▶ Computer-Aided Control Systems Design: Introduction and Historical Overview) have been developed in order to help the user in their design (starting from the identification of the process) by taking into account the different control requirements in a given application (Guzman et al. 2008). In this context, all the additional functionalities can be considered, as well as more complex control architectures where, in any case, the PID control is still the basic element (▶ Control Structure Selection, ▶ Control Hierarchy of Large Processing Plants: An Overview).

Summary and Future Directions

PID controllers are the most employed controllers in industry, and the knowledge about their use is well established, with the presence of many effective tuning and automatic tuning techniques. Despite this, PID controllers are still being developed under many points of view. For example, design methodologies for more complex control schemes (like cascade control or control of multivariable systems with or without the use of a decoupling strategy) can be improved. Further, the advancement of the technologies poses new problems that need to be addressed. For example, the use of wireless sensors and actuators calls for event-based PID controllers whose design should take into account the asynchronous sampling. The availability of faster and faster microprocessors has also stimulated an increasing interest in fractional-order PID controllers which allows a more flexible design at the expense of an increment of the complexity.

Recommended Reading

Basic concepts of PID controllers can be found in almost every book on process control. For a detailed treatment, see (Åström and Hägglund 2006) where all the methodological as well as technological aspects are covered. An excellent collection of tuning rules can be found in O'Dwyer (2006). More advanced topics can be found in Tan et al. (1999), Yu (2006), Johnson and Moradi (2005), Knospe (2006), Visioli (2006), Wang et al. (2008), Visioli and Zhong (2010), and Vilanova and Visioli (2012).

Cross-References

▶ Autotuning
▶ Computer-Aided Control Systems Design: Introduction and Historical Overview
▶ Control Hierarchy of Large Processing Plants: An Overview
▶ Controller Performance Monitoring
▶ Control Structure Selection

Bibliography

Åström KJ, Hägglund T (2004) Revisiting the Ziegler–Nichols step response method for PID control. J Process Control 14:635–650

Åström KJ, Hägglund T (2006) Advanced PID control. ISA Press, Research Triangle Park

Guzman JL, Åström KJ, Dormido S, Hägglund T, Berenguel M, Piguet Y (2008) Interactive learning modules for PID control. IEEE Control Syst Mag 28:118–134

Johnson MA, Moradi MH (eds) (2005) PID control – new identification and design methods. Springer, London

Knospe C (ed) (2006) PID control. IEEE Control Syst Mag 26(1):30–31. Special section

Nichols NB, Ziegler JG (1942) Optimum settings for automatic controllers. Trans ASME 64:759–768

O'Dwyer A (2006) Handbook of PI and PID tuning rules. Imperial College Press, London

Skogestad S (2003) Simple analytic rules for model reduction and PID controller tuning. J Process Control 13:291–309

Tan KK, Wang Q-G, Hang CC, Hägglund T (1999) Advances in PID control. Springer, London

Vilanova R, Visioli A (eds) (2012) PID control in the third millennium: lessons learned and new approaches. Springer, London

Visioli A (2006) Practical PID control. Springer, London

Visioli A, Zhong Q-C (2010) Control of integral processes with dead time. Springer, London

Wang Q-G, Ye Z, Cai WJ, Hang CC (2008) PID control for multivariable processes. Springer, London

Yu CC (2006) Autotuning of PID controllers: a relay feedback approach. Springer, London

Pilot-Vehicle System Modeling

Alexander Efremov
Moscow Aviation Institute, Moscow, Russia

Abstract

The main types and variables of pilot-aircraft systems and pilot control response characteristics are considered. The basic regularities of pilot behavior exposed in closed-loop systems are briefly discussed. Different types of models of pilot behavior are reviewed including classical models (McRuer's and structural) and an optimal control model.

Keywords

Crossover pilot model; Describing function; Manual control; Pilot behavior; Pilot optimal control model Remnant spectral density; Structural model

Introduction

Modern flight control and navigation systems are characterized by two features: (1) they employ fly-by-wire controls and (2) they introduce extensive automation support into the cockpit, ranging from complex augmented flight control systems in manual control modes to powerful flight management computers and autopilots that assume responsibility for most flight control tasks (and which may operate the aircraft in ways that are difficult for pilots to monitor and understand). These modern systems leave the pilot in a supervisory control mode most of the time. Consequently, crew members monitor, supervise, plan, and, in essence, serve as information managers. The level of supervisory control tasks can be different from conventional command control in which the operator issues auto-pilot commands ("set altitude", "set airspeed," etc.) and task-level control in which the operator issues commands such as "line formation," "trail formation," etc. Although civilian pilots have experience flying their aircraft manually, they are seldom in active, direct control of the aircraft. However, if a failure or unexpected upset occurs, they are required to assume control immediately. As for military pilots, they (especially fighter pilots) use manual control in the majority of piloting tasks.

The effective use of manned flight vehicles has always required a satisfactory match of vehicle characteristics (which include vehicle dynamics, control manipulators, displays) with the human pilot's characteristics as a flight controller. The provision of proper vehicle handling qualities by the flight control system and display and manipulator design has often posed serious problems which the vehicle system engineer must solve.

Their solutions require the knowledge of mutual interactions between the pilot and the vehicle. The understanding of such interactions requires a mathematical theory which can be used to explain known findings and to predict new ones. For handling qualities, such theory is based on the methods of control engineering and treats the pilot-vehicle system as a closed-loop (in general, a multiloop) entity. The sine qua non of the theory is a model of pilot dynamic characteristics

P
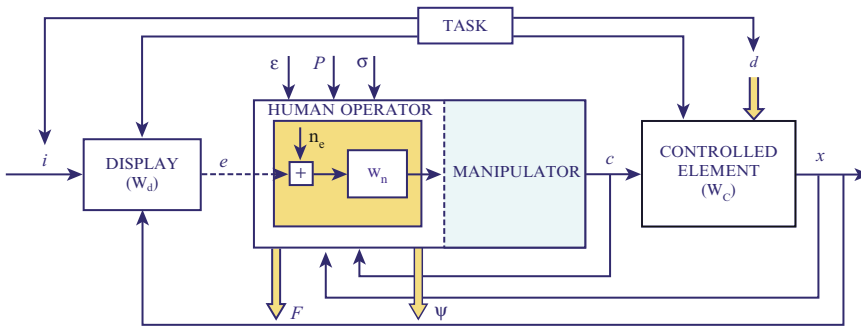
Pilot-Vehicle System Modeling, Fig. 1 Pilot-aircraft system

in a form suitable for application using relatively conventional control engineering techniques. An adequate description of a pilot's dynamics response characteristics is not easily obtained because of the pilot's inherent adaptability and capacity for learning.

Main Variables of the Pilot-Aircraft System

The pilot-aircraft manual control system, shown in Fig. 1, is characterized by a number of variables. The main group of these variables is the so-called task variables which comprise all the system inputs (command inputs $i(t)$, disturbances $d(t)$) and control system elements (display, manipulators, and controlled element dynamics, which is defined by the aircraft frame and flight control system dynamics).

A specific feature of pilot-aircraft systems is the dependence of the piloting task on the task variables. For different piloting tasks, these variables or their parameters differ too. Stability of the closed-loop system is always a necessary, though not sufficient, criterion for the control strategy. Consequently, the pilot's dynamics are profoundly affected by the display and controlled element dynamics, because his response must be adapted to provide the necessary loop stability and accuracy. The characteristics of the other task variables $(i(t), d(t))$, related to the mission and control strategy, also exert direct influence on the pilot dynamics, although their effects are more

in the nature of adjustment and emphasis than of changes in fundamental form.

These variables constitute an enormous range of possible conditions and piloting tasks. In addition to the task variables, the other groups of variables–procedural (p–instructions, training schedule order of presentation of trials etc.), environmental (ε–illumination, vibration, temperature, and so forth), pilot centered (σ–physical condition, motivation etc.)—have less influence on pilot-aircraft system features.

Types of Pilot-Aircraft Systems

The structure of the pilot-aircraft system depends on the piloting task. Some tasks (for example, the pitch tracking task) can be interpreted with the help of the single loop compensatory block diagram. In that case the pilot perceives only the error signal, $y(t) = e(t) = i(t) - x(t)$, and control $c(t)$. Figure 1 is the pilot pitch control command. The other tasks require more complicated descriptions. For example, the landing task is a multiloop compensatory task, where the inner loop closed by the pilot is the pitch control loop. Some piloting tasks are multichannel control tasks, in which the pilot perceives several visual stimuli (for example pitch angle and bank angle) and generates commands in several channels too. Pilots also perceive stimuli of different sensing modalities (visual, vestibular, kinesthetic). In cases where these influence his actions, the multimodality of the pilot-aircraft system has to be analyzed.

A great many past experiments in which human dynamic measurements were taken have been conducted for investigation of compensatory tracking tasks. Some practical piloting tasks (e.g., aim-to-aim tracking in case when the target flies against a background of clouds) correspond to pursuit conditions. In that case, the pilot perceives the information about the error signal $e(t)$ and the input signal $i(t)$.

In many piloting tasks the single loop compensation system defines the main features of more complicated types of pilot-aircraft systems and its flying qualities. Therefore, this type of the system has been investigated in more depth.

Pilot Control Response Characteristics

The most obvious aspect of human dynamic behavior in a manual control task is the pilot's control actions within that task. When the key variables are fixed and the signals in the control loop are approximately time stationary over an interval of interest, the pilot-vehicle system can be presented as a quasi-linear system. In that case, the pilot response can be presented by two components: the pilot-describing function, $W_p(j\omega)$, taking into account the linear portion of pilot response on the stimulus $e(t)$, and remnant $n_e(t)$, which takes into account all nonlinear, nonstationary effects of pilot behavior (Fig. 2).

In the majority of piloting tasks $n_e(t)$ is a stationary process characterizing the remnant spectral density $S_{n_e n_e}(\omega)$ (McRuer and Krendel 1974). The pilot control response characteristics $W_p(j\omega)$ and $S_{n_e n_e}(\omega)$ depend explicitly on the task variables (McRuer and Jex 1967; McRuer et al. 1968). In much experimental research, the technique for identification of these characteristics was based on the use of an input signal consisting of the sum of non-harmonically-related sine waves with cut off frequency ω_i at 1.5, 2.5, and 4 rad/s and different controlled element dynamics (Allen and Jex 1972; Magdaleno 1972; Shirley 1969).

In addition to control response, other types of pilot's responses also characterize his behavior: physiological (F) and psychophysiological ψ responses (Fig. 1). For one of the psychophysiological response characteristics, the pilot opinion rating (PR) is widely used in experimental investigations as well as for the measurement of pilot control response. Pilot opinion ratings are defined by specialized scales (e.g., the Cooper-Harper scale (Cooper and Harper 1969)).
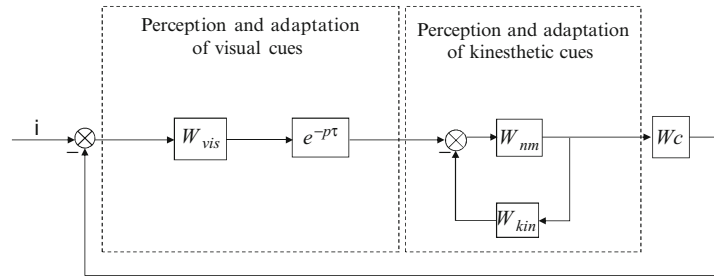
Modeling Pilot Behavior in Manual Control

Experimental investigations have demonstrated a specific regularity: for a variety of forcing functions and controlled elements the slope of the
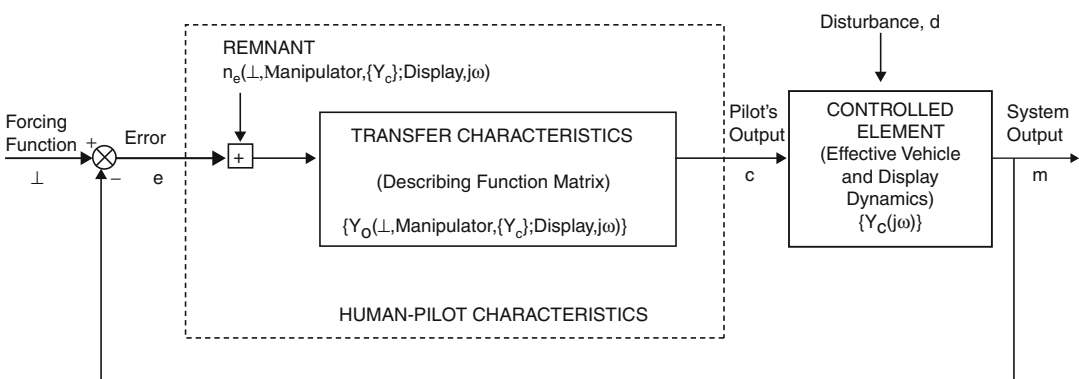
Pilot-Vehicle System Modeling, Fig. 2 Quasi-linear paradigm for the human pilot

Pilot-Vehicle System Modeling, Fig. 3 Pilot structural model

open-loop describing function $|W_{OL}(j\omega)|$ vs frequency was *unity, i.e., −20 dB/dec* in the region of the crossover frequency ω_c (McRuer and Jex 1967). This observation has led to the conclusion that near ω_c, $W_{OL}(j\omega)$ can be presented by the "crossover model" (McRuer and Jex 1967)

$$W_{OL}(j\omega) = W_p(j\omega) \cdot W_C = \frac{\omega_c}{j\omega} e^{-j\omega\tau_e}$$

This model has two parameters:

$$\omega_c = \omega_{co}(\omega_c) + \Delta\omega(\omega_i)$$
$$\tau_e = \tau_o(\omega_c) + \Delta\tau(\omega_i)$$

For the controlled element dynamics $W_C = \frac{K}{s(Ts+1)}$, the increase of constant T leads to an increase of τ_0 and a decrease of ω_{C0}. The empirical dependences of $\Delta\omega_c$ and $\Delta\tau_e$ on ω_i obtained for the rectangular form of input spectrum are the following: $\Delta\omega = 0.18\omega_i$, $\Delta\tau = -0.07\omega_i$.

McRuer proposed several modifications of the open-loop system crossover and pilot describing function models (McRuer and Krendel 1974). One of the simplest ones (used widely in many researches) which might be recommended for description of pilot-aircraft system characteristics in the crossover frequency range is the following

$$W_p(j\omega) = K_p \frac{T_L j\omega + 1}{T_1 j\omega + 1} e^{-j\omega\tau_e}$$

The selection of the parameters K_p, T_L, and T_I is carried out by using "adjustment rules" so that the closed-loop system conforms to experimental frequency response characteristics. These adjustment rules reflect the main features of pilot behavior – adaptation and optimization.

A more complicated model of pilot describing function ("structural model") was offered by R. Hess (1979, 1984). It takes into account the additional inner loop generated by the pilot as a result of his response to the kinesthetic cue (Fig. 3). The modification of this model (Efremov and Tjaglik 2011) demonstrated good agreement with the pilot describing function as measured in experiments. One of the features of this modified model is the criterion used for the parameter optimization: $I = \min[\sigma_e^2]$ or $I = \min[\sigma_e^2 + \beta\sigma_n^2]$. This procedure requires the knowledge of the pilot remnant spectral density. For the single loop system, such a model was developed by Levison et al. (1969).

$$S_{n_e n_e}(\omega) = 0.01\pi \frac{\sigma_e^2 + \sigma_e^2 T_L^2}{1 + T_L^2 \omega^2}$$

In the limited number of researches, the classic approach to pilot modeling considered above was used for more complicated types of the pilot-aircraft system, when the pilot perception of motion cues was taken into account (multimodality system (Hess 1990)) or for a case of the multiloop pilot-aircraft system (Stapleford et al. 1967).

A different approach to pilot behavior modeling was developed by Kleiman et al. (1970). It is based on the modern optimal control theory and assumes that the pilot's goal is to minimize the cost function:

$$I = \lim_T \frac{1}{T} \int_0^T (xQx^T + uQ_c u^T + \dot{u}G_c\dot{u}^T)dt$$

The model takes into account the main pilot limitation parameters: time delay in perception,

Pilot-Vehicle System Modeling, Fig. 4 Optimal pilot model
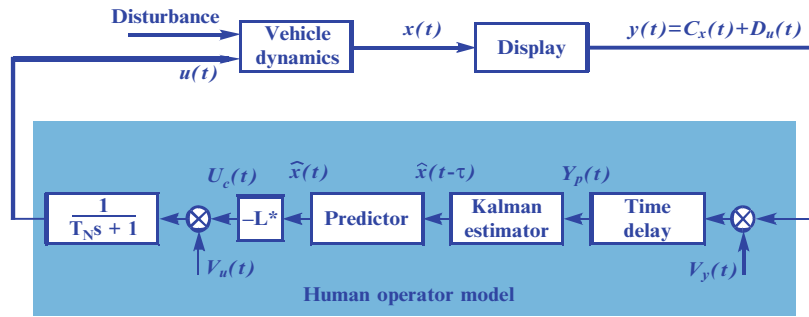
the observation and motor noises, and the neuro-muscular dynamics.

The predictive part of the model consists of the optimal controller ($-L^*$), Kalman filter and predictor (Fig. 4). The software for definition of these elements allows the use of this model for the different types of the pilot-aircraft systems.

The classical and optimal pilot behavior models have been applied widely for different manual control tasks: the development of alternative criteria for flying qualities (Efremov et al. 1998; Neal and Smith 1971), the flight control system (Schmidt 1979) and display design (Klein and Clement 1973), the analysis of reasons for pilot-induced oscillation (McRuer 1997) and the development of means for its suppression (Efremov 1995), and many others.

In some of the researches, attempts have been made to find the relationship between the parameters of the closed-loop system, pilot control response characteristics, and pilot opinion ratings. The technique developed in these researches is called the "paper pilot technique" (Anderson 1970). The following modification of this technique has enabled a close match between the results of mathematical modeling (PR, T_L, accuracy, etc.) of the different types of the pilot-aircraft system and the results of experimental investigations (Efremov and Ogloblin 2006).

Summary and Future Directions

Pilot behavior has been studied extensively for single-loop stationary manual control tasks. Two approaches to the mathematical modeling of the pilot behavior have been developed: classical and optimal control. Both of them have produced good agreement with experimental results. The discussed models describe one of the main features of the pilot adaptation – "parameter adaptation", when a change of any task variable causes a change of human operator control response characteristics. Only a limited number of experimental investigations have been carried out for more complicated cases: multiloop and multimodality pilot-aircraft closed-loop systems. Broader investigations are necessary in the future to obtain accurate pilot mathematical models for these cases. Future investigation in pilot behavior modeling area is also necessary for better formulations of other aspects of pilot adaptation:

– "Structural adaptation", when the pilot selects the loops and the best type of behavior (compensatory, pursuit, etc.) appropriate for the different task variables and, in the case of the flight control system, changes in dynamics.

– "Goal adaptation", when a change of the piloting task or a failure in the controlled element dynamics is accompanied by a change of the goals.

Other future directions in pilot modeling are the development of models to predict the results in the case of sharp changes of controlled element dynamics, to optimize the controlled element dynamics, to define the relationship between the pilot control response characteristics and his opinion rating in different piloting tasks, to get new criteria for the handling qualities, prediction of pilot-induced oscillations, and to solve many other manual control problems.

Cross-References

Bibliography

Allen RW, Jex H (1972) A simple Fourier analysis technique for measuring the dynamic response of manual control systems. IEEE Trans Syst Man Cybern SMC-2(5):638–643

Anderson RO (1970) A new approach to the specification and evaluation of flying qualities. AFFDL-TR-69-120, Wright-Patterson AFB, Ohio, Air Force Flight Dynamics Lab, June 1970

Cooper GE, Harper RP (1969) The use of pilot rating in the evaluation of aircraft handling qualities. NASA TN-D-5153. Moffett Field, CA, NASA Ames Research Center, Apr 1969

Efremov AV (1995) Development and application of the methods for pilot aircraft system research to the manual control tasks of modern vehicles. In: AGARD conference proceedings No. 556 Dual usage in military and commercial technology in guidance and control, Oct 1995

Efremov AV, Ogloblin AV (2006) Progress-in-the-loop investigations for flying qualities prediction and evaluation. ICAS Congress, Hamburg, Sept 2006

Efremov AV, Tjaglik MS (2011) The development of perspective displays for highly precise tracking tasks. In: Holzapfel F, Theil S (eds) Advances in aerospace guidance, navigation and control. Springer-Verlag, Berlin/Heidelberg, pp 163–174

Efremov AV, Ogloblin AV, Predtechensky AN, Rodchenko VV (1992) Pilot as a dynamic system. Mashinostroenije, Moscow, pp 1–343

Efremov AV, Ogloblin AV, Koshelenko AV (1998) Evaluation and prediction of aircraft handling qualities. A collection of technical Papers AIAA Atmospheric Flight Mechanics Conference and Exhibit. AIAA – 98 – 4145, Boston, 10–12 Aug 1998

Hess R (1979) Structural model of the adaptive human behavior. J Guid Control 3(5):416–423

Hess R (1984) The effects of time delay on systems subject to manual control. J Guid Control Dyn 7:165–174

Hess R (1990) A model of human use of motion cues. J Guid Control Dyn 13(3):476–486

Kleiman D, Baron S, Levison W (1970) An optimal control model of human response, parts 1,2. Automatica 6(3):357–369

Klein R, Clement W (1973) Application of manual control display theory to the development of flight director systems for STOL aircraft AFFDL-72-152

Levison W, Baron S, Kleiman D (1969) A model for controller remnant. IEEE Trans MMS-10(4):101–108

Magdaleno RE (1972) Serial segments method for measuring remnant. IEEE Trans Syst Man Cybern SMC-2(5):674–678

McRuer DT (1997) Aviation safety and pilot control understanding and preventing unfavorable pilot-vehicle interactions. National Academy Press, Washington, DC

McRuer DT, Jex HR (1967) A review of quasilinear pilot models. IEEE Trans HFE-8(3):231–249

McRuer DT, Krendel ES (1974) Mathematical models of human pilot behavior. AGARDograph 188:1–72

McRuer DT, Hofmann LG, Jex HR et al (1968) New approaches to human pilot/vehicle dynamic analysis. AFFDL-TR-67-150

Neal TP, Smith RE (1971) A flying qualities criterion for the design of a fighter flight control systems. J Aircraft 8(10):803–809

Schmidt D (1979) Optimal flight control synthesis via pilot modeling. J Guid Control Dyn 4(2):308–312

Shirley R (1969) Application of modified fast Fourier transform to calculate human operator describing functions. IEEE Trans Man-Machine Syst MMS-10(4):140–144

Stapleford R, Ashkenas J et al (1967) Analysis of several handling quality topics pertinent to advanced manned aircraft. AFFDL-TR-67-2, Wright-Patterson ASB, Ohio, Air Force Flight Dynamics Lab, June 1967

PLC

Polynomial/Algebraic Design Methods

Vladimír Kučera
Faculty of Electrical Engineering, Czech Technical University of Prague, Prague, Czech Republic

Abstract

Polynomial techniques have made important contributions to systems and control theory. Algebraic formalism offers several useful tools for control system design. In most cases, control systems are designed to be stable and to meet additional performance specifications, such as optimality or robustness. The basic tool is a

parameterization of all controllers that stabilize a given plant. Optimal or robust controllers are then obtained by an appropriate selection of the parameter. An alternative tool is a reduction of controller synthesis to a solution of a polynomial equation of specific type. These two polynomial/algebraic approaches will be presented as closely related rather than isolated alternatives.

Keywords

Controller synthesis; Linear systems; Polynomial equation approach to control system design; Youla-Kučera parameterization of stabilizing controllers

Stabilizing Controllers

The majority of control problems can be formulated using the diagram shown in Fig. 1. Given a plant S, determine a controller R such that the feedback control system is stable and satisfies some additional performance specifications, such as reference tracking, disturbance attenuation, optimality, or robustness.

Suppose that the plant and the controller are linear time-invariant single-input single-output continuous-time systems with real *rational* transfer functions S and R, respectively. Stability is understood as the *input-output stability*, i.e., whenever the exogenous inputs δ and ρ are essentially bounded in amplitude, so too are the output signals μ and η (hence also ε and ν).

It is natural to separate the design task into two consecutive steps: (1) stabilization and (2) achievement of additional performance specifications. To do this, all solutions of the first step, i.e., *all controllers that stabilize the given plant*, must be found.

How can one characterize such controllers? Denote H_s the reference-to-error transfer function (sometimes called the sensitivity function) and H_c the disturbance-to-control transfer function (the so-called complementary sensitivity function) in the closed-loop control system, namely,

$$H_s = \frac{1}{1 + SR}, \quad H_c = \frac{SR}{1 + SR}.$$

Now suppose that S can be expressed as the ratio of two coprime polynomials, $S = b/a$, and that the controller has alike form, $R = q/p$. Then the two closed-loop transfer functions can be written as

$$H_s = a \, \frac{p}{ap + bq} := aX,$$

$$H_c = b \, \frac{q}{ap + bq} := bY$$

Consequently, if R stabilizes S, then the rational functions X and Y are bound to be stable. These functions cannot be arbitrary, however, since $H_s + H_c = 1$. The stability equation follows as

$$aX + bY = 1.$$

Any stabilizing controller for S can be expressed as $R = Y/X(= q/p)$, where X and Y are a stable rational solution pair of the stability equation. This solution can be expressed in parametric form:

$$X = x + bW, \quad Y = y - aW,$$

furnishing in turn an explicit parameterization of the set of all stabilizing controllers for S:

$$R = \frac{y - aW}{x + bW},$$

known as the *Youla-Kučera parameterization*. Here x and y are any polynomials satisfying the Bézout equation $ax + by = 1$, while W is a free parameter ranging over the set of stable real rational functions such that $x + bW$ is not identically zero.
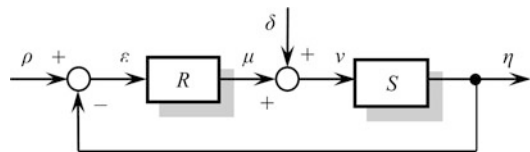
Polynomial/Algebraic Design Methods, Fig. 1 Feedback control system

Example 1 Consider an integrator plant $S(s) = 1/s$. The Bézout equation admits a solution $x = 0$, $y = 1$ so that the set of all stabilizing controllers for S is given by

$$R(s) = \frac{1 - sW}{W}$$

for any stable real rational $W \neq 0$.

The parameter

$$W(s) = \frac{1}{s + 1}$$

yields $R = 1$, a proportional gain controller. The parameter

$$W(s) = \frac{s}{s^2 + s + 1}$$

results in a proportional-integral controller

$$R(s) = 1 + \frac{1}{s}.$$

Taking $W = 1$ leads to the stabilizing controller $R(s) = 1 - s$. The feedback system is stable, but it has a pole at $s = \infty$.

Additional Performance Specifications

There is a simple formula that generates all the stabilizing controllers for a given plant. Using this formula, we can obtain a parameterization of all stable closed-loop transfer functions that can be obtained by stabilizing a given plant. The bonus is that the parameterization is *affine* in the free parameter W. In contrast, the controller R appears in a nonlinear fashion:

$$\begin{bmatrix} v \\ y \end{bmatrix} = \frac{1}{1 + SR} \begin{bmatrix} 1 & R \\ S & SR \end{bmatrix} \begin{bmatrix} d \\ r \end{bmatrix}$$

$$= \begin{bmatrix} a(x + bW) & a(y - aW) \\ b(x + bW) & b(y - aW) \end{bmatrix} \begin{bmatrix} d \\ r \end{bmatrix}.$$

As R and W are in a one-to-one correspondence, it is convenient to use W in lieu of R in the

design process and calculate R subsequently. Thus, the parameterization of all stabilizing controllers makes it possible to separate the design process into two steps: the determination of all stabilizing controllers and the selection of the parameter that achieves the remaining design specifications. The extra benefit is that both tasks are linear.

Asymptotic Properties

Asymptotic properties of control systems can easily be accommodated in the sequential design procedure. These include the elimination of an offset due to step references, the ability of system output to follow a class of reference signals, or the asymptotic elimination of specific disturbances.

In Fig. 1, asymptotic *reference tracking* means that the output η follows the reference ρ as time approaches infinity, which is to say that the error ε approaches zero for large times. On the other hand, we speak of asymptotic *disturbance elimination* if the effect of the disturbance δ decreases at the output η for increasing time. In terms of Laplace transforms, $\varepsilon = H_s\rho$ and $\eta = SH_s\delta$ are to be *stable* rational functions.

Example 8.1 Consider the plant $S(s) = 1/(s + 1)$. The Bézout equation admits a solution $x = 0$, $y = 1$. The set of all stabilizing controllers for S is

$$R(s) = \frac{1 - (s + 1)W}{W}$$

for any stable real rational $W \neq 0$. The achievable sensitivity transfer functions are $H_s = (s + 1)W$.

To track a step reference, $\rho = 1/s$, we must take $W = sW_1$ for any stable rational $W_1 \neq 0$. To eliminate a sinusoidal disturbance, $\delta = s/(s^2 + \omega^2)$, we constrain the parameter as $W = (s^2 + \omega^2)W_2$ for any stable rational $W_2 \neq 0$. To meet both requirements, we simply take $W = s(s^2 + \omega^2)W_3$ for any stable rational $W_3 \neq 0$, say $W = s(s^2 + \omega^2)/(s + 1)^4$.

The resulting controller is

$$R(s) = \frac{3s^3 + (6 - \omega^2)s^2 + (4 - \omega^2)s + 1}{s(s^2 + \omega^2)}.$$

The controller obtained in Example 8.1 demonstrates the internal model principle: the unstable modes to be followed or eliminated must be generated by the controller unless they are present in the plant.

H$_2$ Optimal Control

The sequential design procedure will be further illustrated on the design of *linear-quadratic optimal* controllers. Given a plant with transfer function $S = b/a$, the task is to find a controller that stabilizes the control system of Fig. 1 while minimizing the H_2 norm of some closed-loop transfer function, say of the complementary sensitivity function H_c.

The H_2 norm is defined for any strictly proper rational function G analytic on the imaginary axis as

$$\| G \|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 \, d\omega} \,.$$

The set of complementary sensitivity functions that can be achieved in the stabilized control system is

$$H_c = b(y - aW) \,,$$

where W is a free stable rational parameter. The parameter will be selected so as to minimize the H_2 norm of H_c.

Let $\alpha\beta$ be a polynomial defined by keeping the stable (in Re $s < 0$) zeros of ab while replacing the unstable (in Re $s \geq 0$) ones with their negative values. Then $ab/\alpha\beta$ is inner (or all pass) and

$$\| H_c \|_2 = \left\| \frac{\alpha\beta}{ab} H_c \right\|_2 = \left\| \frac{\alpha y \beta}{a} - \alpha W \beta \right\|_2 .$$

Consider the decomposition

$$\frac{\alpha y \beta}{a} = r + \frac{q}{a}$$

where r is a polynomial and q/a is strictly proper. With this decomposition,

$$\| H_c \|_2^2 = \left\| \frac{q}{a} \right\|_2^2 + \| r - \alpha W \beta \|_2^2$$

because q/a and $r - \alpha W\beta$ are orthogonal and thus the cross-terms contribute nothing to the norm. The last expression is a complete square whose first part is independent of W. Hence the minimizing parameter is $W = r/\alpha\beta$, and if it is indeed stable and admissible, it defines the unique optimal controller. Otherwise, no optimal controller exists.

The consequent minimum norm equals

$$\min_W \| H_c \|_2 = \left\| \frac{q}{a} \right\|_2 .$$

Example 8.2 To illustrate, consider the plant $S(s) = 1/(s - 1)$. The class of all stabilizing controllers for S is found to be

$$R(s) = \frac{1 - (s - 1)W}{W}$$

for a free stable rational parameter $W \neq 0$. The complementary sensitivity transfer function is

$$H_c(s) = 1 - (s - 1)W.$$

Now $\alpha = s + 1, \beta = 1$ and the polynomial part of

$$\frac{\alpha y \beta}{a} = \frac{s + 1}{s - 1} = 1 + \frac{2}{s - 1}$$

is $r = 1$. Thus H_c attains minimum H_2 norm for

$$W(s) = \frac{1}{s + 1}$$

and the corresponding optimal controller is $R(s) = 2$.

The optimal complementary sensitivity function is

$$H_c(s) = \frac{2}{s + 1}$$

and $\| H_c \|_2 = \sqrt{2}$.

Robust Stabilization

The notion of robust stability addresses stabilization of plants subject to modeling errors, when the actual plant may differ from the nominal model, using a fixed controller. The ultimate goal is to stabilize the actual plant. The actual plant is

unknown, however, so the best one can do is to stabilize a large enough set of plants.

Thus the basis technique to model plant uncertainty is to model the plant as belonging to a set. Such a set can be either structured – for example, there is a finite number of uncertain parameters – or unstructured: the frequency response lies in a set in the complex plane for every frequency. The unstructured uncertainty model is more important for several reasons. On the one hand, it is well suited to represent high-frequency modeling errors, which are generically present and caused by such effects as infinite-dimensional electromechanical resonance, transport delays, and diffusion processes. On the other hand, the unstructured model of uncertainty leads to a simple and useful design theory.

The unstructured set of plants is usually constructed as a neighborhood of the nominal plant, with the uncertainty represented by additive or multiplicative perturbations. The size of the neighborhood is measured by a suitable norm, most common being the H_∞ norm that is defined for any rational function G analytic on the imaginary axis as

$$\| G \|_\infty = \sup_\omega | G(j\omega) | \ .$$

Let us illustrate the design for *robust stability under unstructured norm-bounded multiplicative perturbations*. Consider a nominal plant with transfer function S and its neighborhood S_Δ defined by

$$S_\Delta := (1 + F\Delta)S$$

where F is a fixed stable rational function and Δ is a variable stable rational function such that $\| \Delta \|_\infty \leq 1$.

The idea behind this uncertainty model is that $F\Delta$ is the normalized plant perturbation away from 1:

$$\frac{S_\Delta}{S} - 1 = F\Delta \ .$$

Hence if $\| \Delta \|_\infty \leq 1$, then for all frequencies ω

$$\left| \frac{S_\Delta(j\omega)}{S(j\omega)} - 1 \right| = | F(j\omega) |$$

so that $| F(j\omega) |$ provides the uncertainty profile while Δ accounts for phase uncertainty.

Now suppose that R is a controller that stabilizes the nominal plant S. Applying the small gain theorem, R is seen to stabilize the entire family of plants S_Δ if and only if

$$\| H_c F \|_\infty < 1.$$

This is a necessary and sufficient condition for robust stabilization of the nominal plant S.

The set of all stabilizing controllers for $S = b/a$ is described by the formula

$$R = \frac{y - aW}{x + bW}$$

where $ax + by = 1$ and W is a free stable rational parameter. The robust stability condition then reads

$$\| b(y - aW)F \|_\infty < 1.$$

Any stable rational W that satisfies this inequality then defines a robustly stabilizing controller R for S. In case W actually minimizes the norm, one obtains the best robustly stabilizing controller.

Example 8.3 Consider a plant with the transfer function

$$S_\tau(s) = \frac{s + 1}{s - 1} e^{-\tau s}$$

where the time delay τ is known only to the extent that it lies in the interval $0 \leq \tau \leq 0.2$. The task is to find a controller that stabilizes the uncertain plant S_τ. The time-delay factor $e^{-\tau s}$ can be treated as a multiplicative perturbation of the nominal plant
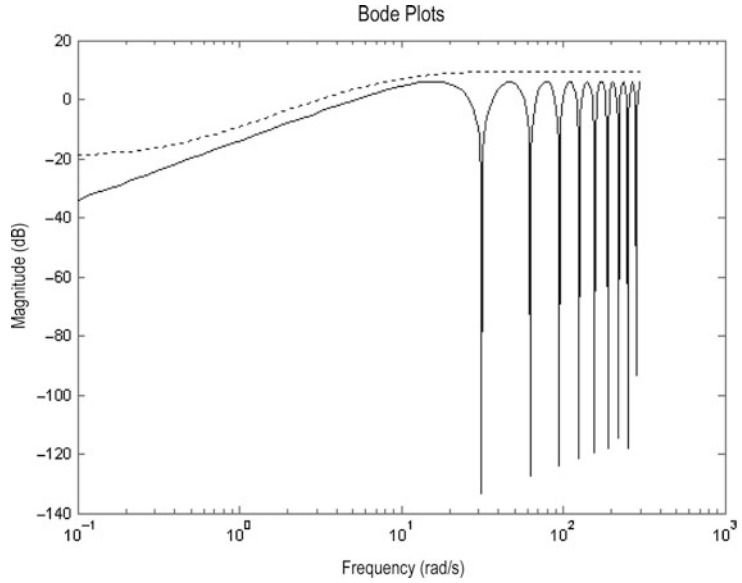
$$S(s) = \frac{s + 1}{s - 1}$$

by embedding S_τ in the family

$$S_\Delta := (1 + F\Delta)S$$

where Δ ranges over the set of stable rational functions such that $\| \Delta \|_\infty \leq 1$. To do this, F

**Polynomial/Algebraic
Design Methods, Fig. 2**
Bode plots of F (*dotted*)
and $e^{-0.2s} - 1$ (*solid*)

should be chosen so that the normalized perturbation satisfies

$$\left| \frac{S_\Delta(j\omega)}{S(j\omega)} - 1 \right| = \left| e^{-j\omega\tau} - 1 \right| \le |F(j\omega)|$$

for all ω and τ. A little time with the Bode magnitude plot shows that a suitable uncertainty profile is

$$F(s) = \frac{3s + 1}{s + 9}.$$

Figure 2 is the Bode magnitude plot of this F and $e^{-\tau s} - 1$ for $\tau = 0.2$, the worst value.

The task of stabilizing the uncertain plant S_τ is thus replaced by that of stabilizing every element in the set S_Δ, that is to say, by robustly stabilizing the nominal plant S with respect to the multiplicative perturbations defined by F.

The set of all stabilizing controllers for S is found to be

$$R(s) = \frac{0.5 - (s - 1)W}{-0.5 + (s + 1)W}$$

where $W \ne 0.5/(s + 1)$ is any stable rational parameter. The robust stability condition reads

$$\| P - QW \|_\infty < 1$$

where

$$P(s) = 0.5(s + 1)\frac{3s + 1}{s + 9},$$

$$Q(s) = (s - 1)(s + 1)\frac{3s + 1}{s + 9}.$$

Since Q has one unstable zero at $s = 1$, it follows from the maximum modulus theorem that the minimum of the H_∞ norm taken over all stable rational functions W is $P(1) = 0.4 < 1$ and this minimum is achieved for

$$W(s) = \frac{P(s) - P(1)}{Q(s)} = \frac{1}{10}\frac{15s + 31}{(s + 1)(3s + 1)}.$$

Thus, the robust stability condition is satisfied, and the corresponding best robustly stabilizing controller is

$$R(s) = \frac{2}{13}\frac{s + 9}{s + 1}.$$

Polynomial Equation Approach

In order to determine the set of all stabilizing controllers for a given plant, it is enough to determine one particular solution of the Bézout equation. It is therefore plausible that performance

specifications in addition to stability can be met by selecting an appropriate solution of a polynomial equation that is related to the Bézout equation.

The reduction of controller synthesis to solving polynomial equations is referred to as the *polynomial equation approach* to control system design. The equations involved are Diophantine equations of the form

$$ap + bq = d$$

where a, b, and d are given polynomials and p, q are polynomials to be found. Such an equation is solvable for any d if and only if a and b are coprime polynomials. Then, the solution set is given by

$$p = p_0 + bt, \quad q = q_0 - at$$

where p_0, q_0 is a particular solution and t is an arbitrary polynomial.

Such an equation is in fact the pole placement equation. Thus, pole placement is a prototype control problem.

Pole Placement

The requirement of stability places all closed-loop system poles within the left half-plane Re $s < 0$. Very often, however, we wish to allocate the poles to a specific region of the half-plane or to achieve specific pole positions.

Given a plant $S = b/a$, the set of all stabilizing controllers for S is

$$R = \frac{y - aW}{x + bW}$$

where x, y are polynomials such that $ax + by = 1$ and W is a free stable rational parameter. Let $W = w/d$ for a stable polynomial d. Then

$$R = \frac{dy - aw}{dx + bw} := \frac{q}{p}$$

and the closed-loop pole polynomial is given by

$$ap + bq = d(ax + by) = d.$$

Thus W parameterizes all stabilizing controllers for S, the denominator polynomial d of W specifies the positions of the control system poles, and the numerator polynomial w of W represents the remaining degrees of freedom, i.e., parameterizes all stabilizing controllers that assign the specified poles.

Example 8.4 Consider the plant $S(s) = 1/(s-1)$ and the set of stabilizing controllers for S:

$$R(s) = \frac{1 - (s - 1)W}{W}, \quad W \neq 0.$$

Let the desired pole locations be given by the polynomial $d = s^2 + 2s + 1$. This is achieved by putting $W = w/d$ for an arbitrary numerator polynomial $w \neq 0$.

It is to be noted that d specifies the poles at *finite* positions only. Poles at $s = \infty$ will occur whenever R is not proper rational. To avoid this situation, w should be constrained to $w = s + \omega$ for any real ω. Then the set of controllers that achieve the desired pole placement is

$$R(s) = \frac{(3 - \omega)s + (1 + \omega)}{s + \omega}.$$

Alternatively, one can solve the pole placement equation $ap + bq = d$ directly. The solution set is

$$p = t, \quad q = s^2 + 2s + 1 - (s - 1)t$$

and q/p is proper if and only if $t = s + \omega$, ω real.

H₂ Optimal Control

The H_2 optimal control is a special case of pole placement. Indeed, the optimal controller is given by

$$R = \frac{y - aW}{x + bW} = \frac{y - a\frac{r}{\alpha\beta}}{x + b\frac{r}{\alpha\beta}} = \frac{\alpha y\beta - ar}{\alpha x\beta + br} := \frac{q}{p}$$

and

$$ap + bq = a(\alpha x\beta + br) + b(\alpha y\beta - ar)$$
$$= \alpha\beta(ax + by) = \alpha\beta.$$

Thus, the (finite) pole positions of the H_2 optimal control system are given by the pole polynomial $d = \alpha\beta$. The system has no poles at $s = \infty$ as the optimal complementary sensitivity function H_c is strictly proper.

The pole placement equation, however, has more than one solution. Which one is optimal? The one with q/a is strictly proper. It is the solution pair p, q with q having a least degree.

Example 8.5 Let us reconsider Example 8.2. As an alternative, one can solve the Diophantine equation

$$(s - 1)p + q = s + 1$$

for the solution pair p, q such that $q/(s - 1)$ is strictly proper. This yields the least-degree solution pair with respect to q, namely, $p = 1$, $q = 2$. The optimal controller is $R = q/p = 2$.

Summary and Future Directions

The benefits of representing stabilizing controllers by a single parameter include (1) easy accommodation of additional design specifications by selecting an appropriate parameter, (2) all transfer functions in a stabilized system are linear in the parameter (while they are nonlinear in the controller), and (3) the parameter belongs to a smaller set of stable rational functions (while the controller is any rational).

The results presented here for linear time-invariant systems with rational transfer functions can be generalized to extend the scope of the theory to include distributed-parameter systems, time-varying systems, and even nonlinear systems.

The transfer functions of distributed-parameter systems are no longer rational, and coprime factorizations cannot be assumed a priori to exist. The coefficients of time-varying systems are functions of time, and the operations of multiplication and differentiation do not commute. In nonlinear systems, transfer functions are replaced by input-output maps.

Technical assumptions may prevent one from parameterizing the *entire* set of internally stabilizing controllers; still, the subset may be large enough for practical purposes. For many systems of physical and engineering interest, the above difficulties can be circumvented and the algebraic/polynomial approach carries over with suitable modifications.

Cross-References

▶ Control of Linear Systems with Delays
▶ Feedback Stabilization of Nonlinear Systems
▶ H-Infinity Control
▶ H_2 Optimal Control
▶ Linear State Feedback
▶ Robust Synthesis and Robustness Analysis Techniques and Tools
▶ Spectral Factorization
▶ Tracking and Regulation in Linear Systems

Recommended Reading

The use of polynomials, in one way or another, in feedback control system design can be traced back to Newton et al. (1957) and Jury (1958). The authors noted that for a closed-loop system to be stable, H_c must absorb the plant unstable zeros. The plant was assumed to be stable; if this assumption were dropped, H_s would have been found to absorb the plant unstable poles. These conditions are equivalent to polynomial divisibility conditions and hence to the Bézout stability equation, which appears later in Kučera (1974).

The first attempt to use polynomials in an explicit manner is due to Volgin (1962), a student of Tsypkin. He obtained a solution of the pole placement problem through the solution of a polynomial equation, known as the pole placement equation. Åström (1970) published a polynomial equation solution to the minimum variance control problem for minimum-phase plants. The ultimate publication that presents

the polynomial equation approach to multi-input multi-output control system design is Kučera (1979).

The underlying problem in any control system design is that of stability. It is logical to design the control system step by step: stabilization first and then the additional performance specifications. To do this, we need to know any and all stabilizing controllers for the given plant.

This problem was first addressed and solved for finite-dimensional, linear time-invariant systems using transfer function methods; see Larin et al. (1971), Kučera (1975), Youla et al. (1976a,b), and Kučera (1979). A state-space representation of all stabilizing controllers was derived later by Nett et al. (1984).

It took decades to appreciate the importance of the result and come up with applications. The milestones were the observations by Desoer et al. (1980) that the polynomial fraction approach can be extended to linear systems with nonrational transfer functions, as well as the result by Hammer (1985) showing that the approach is applicable to a broad class of nonlinear systems. Further generalizations were obtained by Paice and Moore (1990), Anderson (1998), and Quadrat (2003, 2006).

The parameterization of all controllers that stabilize a given plant was labeled the Youla-Kučera parameterization in Anderson (1998). This result launched an entirely new area of research and has ultimately become a new paradigm for control system design.

Tutorial textbooks on this subject include Vidyasagar (1985), Doyle et al. (1992), and Kučera (2003, 2011). The reader is further referred to the survey papers by Kučera (1993), Anderson (1998), and Kučera (2007).

Advanced and recent applications of the Youla-Kučera parameterization include stabilization under constrained inputs (Henrion et al. 2001), robust stabilization with fixed-order controllers (Henrion et al. 2003), accommodation of time-domain constraints on inputs and outputs (Henrion et al. 2005a), and determination of least-order stabilizing controllers (Henrion et al. 2005b).

Bibliography

Anderson BDO (1998) From Youla-Kučera to identification, adaptive and nonlinear control. Automatica 34:1485–1506

Åström KJ (1970) Introduction to stochastic control theory. Academic, New York

Desoer CA, Liu RW, Murray J, Saeks R (1980) Feedback system design: the fractional representation approach to analysis and synthesis. IEEE Trans Autom Control 25:399–412

Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan, New York

Hammer J (1985) Nonlinear system stabilization and coprimeness. Int J Control 44:1349–1381

Henrion D, Tarbouriech S, Kučera V (2001) Control of linear systems subject to input constraints: a polynomial aproach. Automatica 37:597–604

Henrion D, Šebek M, Kučera V (2003) Positive polynomials and robust stabilization with fixed-order controllers. IEEE Trans Autom Control 48:1178–1186

Henrion D, Tarbouriech S, Kučera V (2005a) Control of linear systems subject to time-domain constraints with polynomial pole placement and LMIs. IEEE Trans Autom Control 50:1360–1364

Henrion D, Kučera V, Molina-Cristobal A (2005b) Optimizing simultaneously over the numerator and denominator polynomials in the Youla-Kučera parametrization. IEEE Trans Autom Control 50:1369–1374

Jury EI (1958) Sampled-data control systems. Wiley, New York

Kučera V (1974) Closed-loop stability of discrete linear single variable systems. Kybernetika 10:146–171

Kučera V (1975) Stability of discrete linear feedback systems. In: Proceedings of the 6th IFAC world congress, Boston, vol 1, pp 44.1

Kučera V (1979) Discrete linear control: the polynomial equation approach. Wiley, Chichester

Kučera V (1993) Diophantine equations in control – a survey. Automatica 29:1361–1375

Kučera V (2007) Polynomial control: past, present, and future. Int J Robust Nonlinear 17:682–705

Kučera V (2003) Parametrization of stabilizing controllers with applications. In: Voicu M (ed) Advances in automatic control. Kluwer, Boston, pp 173–192

Kučera V (2011) Algebraic design methods. In: Levine WS (ed) The control handbook: control system advanced methods, 2nd edn. CRC, Boca Raton

Larin VB, Naumenko KI, Suntsev VN (1971) Spectral methods for synthesis of linear systems with feedback (in Russian). Naukova Dumka, Kiev

Nett CN, Jacobson CA, Balas MJ (1984) A connection between state-space and doubly coprime fractional representations. IEEE Trans Automat Control 29:831–832

Newton G, Gould L, Kaiser JF (1957) Analytic design of linear feedback controls. Wiley, New York

Paice ADB, Moore JB (1990) On the Youla-Kučera parametrization of nonlinear systems. Syst Control Lett 14:121–129

Quadrat A (2003) On a generalization of the Youla-Kučera parametrization. Part I: the fractional ideal approach to SISO systems. Syst Control Lett 50:135–148

Quadrat A (2006) On a generalization of the Youla-Kučera parametrization. Part II: the lattice approach to MIMO systems. Math Control Signal 18:199–235

Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge

Volgin LN (1962) The fundamentals of the theory of controlling machines (in Russian). Soviet Radio, Moscow

Youla DC, Bongiorno JJ, Jabr HA (1976a) Modern Wiener-Hopf design of optimal controllers, part I: the single-input case. IEEE Trans Autom Control 21:3–14

Youla DC, Jabr HA, Bongiorno JJ (1976b) Modern Wiener-Hopf design of optimal controllers, part II: the multivariable case. IEEE Trans Autom Control 21:319–338

Power System Voltage Stability

Costas Vournas
School of Electrical and Computer Engineering, National Technical University of Athens, Zografou, Greece

Abstract

Voltage stability of electric power systems is a challenging topic both theoretically and in practice. This article touches briefly on the main aspects of the problem and highlights theoretical foundations and fundamental methods for voltage stability analysis. The single-load radial system is used to introduce relevant concepts, such as the *PV* curve and the instability mechanism, while the implications for a meshed, multiple-load system are briefly outlined. Some applications to practical problems are briefly enumerated.

Keywords

Introduction

Voltage stability is related to the maximum power transfer in an AC (alternating current) network. In normal conditions, system load demand should never come close to this limit. As, however, electricity demand started swelling after 1970s with an increasingly faster pace, transmission network investments could not follow closely enough. Investment cost in transmission is usually high, and difficulties with environmental constraints and "not in my back yard" mentality of local communities did not make transmission network expansion any easier. Power systems are thus relying for their continuing operation more and more on (reactive power) compensation and automatic controls to maintain transmission capacity of relatively weakening networks.

As a result several instances of voltage instability started to appear in several industrialized countries after the 1980s (Taylor 1994) leading to smaller or larger area blackouts, much to the surprise of the power engineering community that was not prepared to deal with this type of events, in which a usual and expected phase of gradual voltage decline suddenly precipitates to an uncontrollable voltage drop leading to partial or total blackout after a succession of equipment disconnection by protection devices.

In power system engineering practice, voltage drops following load ramping or sudden events, such as equipment loss (line, generator switching, etc.), usually referred to in power engineering literature as contingencies, are calculated by solving a set of nonlinear algebraic equations known as the power flow problem. As these are "steady-state" equations, the dynamic aspect leading to an accelerating, cascading failure is not obvious. One should notice however in the above account the keyword "nonlinear": nonlinear equations at the maximum power transfer limit no longer have a solution. This was and is one of the keys in understanding the voltage stability problem. To take it one step further, close to the loss of solution (loss of equilibrium), a set of dormant (up to this point) dynamics become dominant leading the system to instability. The following sections will explain these notions.
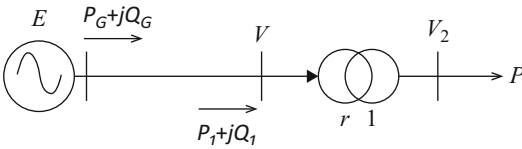
Power System Voltage Stability, Fig. 1 Single-load radial system

Single Generator-Load (Radial) System

Maximum Power Transfer

In any electric network (DC or AC), there is a maximum power that can be transferred between any two nodes. In a two-node radial system, the maximum power transfer coincides with the well-known impedance matching conditions. For a radial AC system, when the load is restricted to a constant power factor, the impedance matching condition is that the source (network) impedance is equal in magnitude to the load impedance.

Consider the radial system of Fig. 1. In this system we assume that the load active power P (and possibly reactive power Q) is fed through a transformer with adjustable tap ratio r (in per unit). The tap is automatically adjusted by a load tap changer (LTC) so as to keep the secondary voltage V_2 within a deadband. We will consider throughout that the LTC is a part of the load.

The simplest case for this radial system is when both the line and transformer are lossless ($P_G = P_1 = P$) and the load is kept to unity power factor ($Q = 0$). The generator is assumed as a constant voltage source E. If we further assume that the transformer leakage reactance is negligible ($Q_1 = Q = 0$ in Fig. 1), the maximum power transfer in this simple case is encountered when the load impedance, as seen from the primary (r^2/G), is equal to the line reactance:

$$X = r^2/G \qquad (1)$$

where the load conductance $G = P/V_2^2$. It can be readily shown that the maximum power in this case is $P_{\max} = E^2/2X$. Note that this is a static condition that is not related to how the load varies with the voltage V_2.

The most popular way of visualizing the maximum power condition is through the *PV* curve of Fig. 2, in which the consumed (transferred) power P is plotted versus the primary (transmission) side voltage V.

In Fig. 2 the nose-shaped solid line is the *network characteristic* corresponding to all possible solution of the network equations for a given P (or V). The maximum power transfer is easily identified as the tip of the curve (point C). Note that *PV* curves can be plotted for any load power factor and line resistance.

Load Dynamics and Voltage Stability

As stated above, maximum power transfer is a static condition based on network equations only. To identify its relation to voltage stability, some form of load dynamics must be introduced. Load dynamics are generally changing the load characteristics so as to adjust load power *consumption* P to a given load *demand* P_o. As a disturbance usually reduces voltage (and thus consumption of a voltage-sensitive load), load dynamics tend to restore the consumption to the pre-disturbance demand.

Load restoration can be continuous, for instance, represented by a time-varying conductance following the ODE:

$$T\dot{G} = \frac{P_o}{V_o^2} - \frac{P}{V_o^2} = \frac{P_o}{V_o^2} - G\left(\frac{V_2}{V_o}\right) \qquad (2)$$

Clearly in this case, the stability condition is that the consumption $P = GV_2^2$ increases with the increase of the load conductance G:

$$\frac{\partial P}{\partial G} > 0 \qquad (3)$$

It is easily verified from Fig. 2 that this condition is met only in the upper part of the *PV* curve before point C, whereas in the lower part, after point C, increased conductance results in lower consumption violating (3). Clearly at C, (3) holds as an equality.

Assuming the load on the secondary side to be a constant admittance, we can distinguish two types of load characteristics in Fig. 2: the

Power System Voltage Stability, Fig. 2 *PV* curve of the radial system
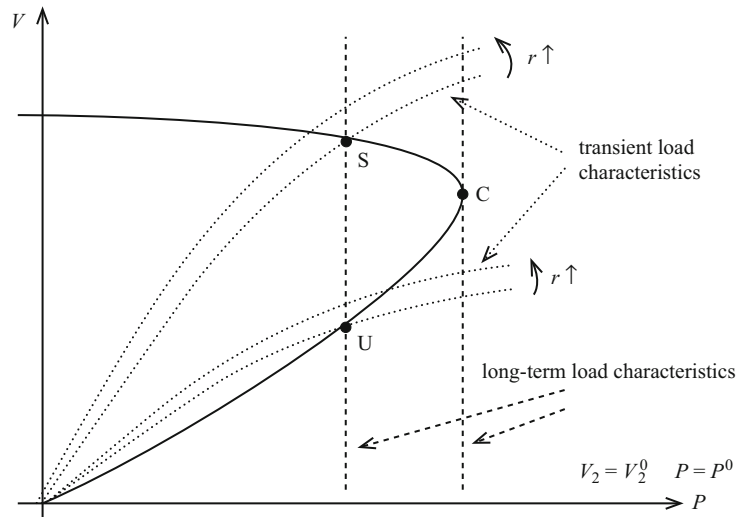
transient (short-term) load characteristic shown with dotted lines corresponds to a specific transformer tap ratio r, whereas the *long-term load characteristic* corresponds to equilibrium conditions where V_2 is within the deadband and approximately equal to V_o and is shown with dashed lines for different load demands.

Load dynamics can also be discrete, e.g., driven by the tap changing transformer of Fig. 1. As the LTC is trying to restore the secondary voltage, it will reduce r when $V_2 < V_o{-}d$ and will increase r when $V_2 > V_o + d$, where d is half of the deadband.

The effect of tap ratio increase in the upper and lower part of the *PV* curve is shown in Fig. 2 (points S and U). In the upper part, increased r will reduce consumption which implies that V_2 is also reduced as expected. In the lower part (point U), increased r will increase consumption indicating an increased V_2 and thus an unstable LTC operation. The stability condition in this case is

$$\frac{\partial V_2}{\partial r} < 0 \qquad (4)$$

Clearly for either discrete or continuous dynamics, at the maximum power point C, a stable and an unstable equilibrium branch come together, leaving no equilibrium points for higher demand. In bifurcation theory this point is known as a saddle-node bifurcation (SNB).

Effect of Generation

The generator behind the constant voltage source E of Fig. 1 supplies the active power consumed by the load (it would cover also active losses, if present). In practice this means that it requires a governor with PI (proportional plus integral) control, as is customary for autonomous systems and a prime mover of the required capacity. The generator also maintains the constant voltage E assumed in the calculations. This requires an automatic voltage regulator (AVR), which can be assumed in this simple example as being also of PI type. The AVR is adjusting the DC rotor (field) current of the synchronous generator so as to maintain the terminal voltage constant.

For a given load, the active and reactive generation $P_G + jQ_G$ required is directly calculated from the network equations. The electromotive force (EMF) corresponding to the field current can then be determined using standard synchronous machine equations and preferably taking into account the saturation of the machine iron core (Van Cutsem and Vournas 1998). It should be noted that due to thermal constraints, it is not possible to exceed a maximum rotor current in continuous operation. This results in a rotor current limit that is enforced by the generator overexcitation limiter (OEL). If loading conditions are such that the OEL is activated, the generator terminal voltage

E cannot be maintained constant, and thus the voltage source *E* has to be replaced by a constant EMF in series with the generator reactance. This leads to a much more restrictive limit for the maximum power transfer.

In power flow calculations, the generator excitation limit is usually represented by a maximum allowable reactive generation $Q_G^{\max}$. When this limit is reached, the reactive generation remains constant, and thus the terminal voltage is allowed to vary, i.e., the generator becomes a PQ bus. Note however that $Q_G^{\max}$ of an actual generator is not constant but depends on terminal voltage and on active generation.

In any case the overexcitation limit of synchronous generators and the resulting limitation of the reactive support they offer is an important factor determining maximum power and thus voltage stability limits. In practice voltage instability is reached only after some critical generators have reached the overexcitation limit.

Voltage Instability Mechanism

Following the preceding discussion, it is possible to describe the mechanism of voltage instability as follows (Van Cutsem and Vournas 1998):

> Voltage instability stems from the attempt of load dynamics to restore power consumption beyond the capability of the combined transmission and generation system.

A voltage instability incident can occur either through a gradual load increase up to the maximum power limit or most commonly following a contingency (or a cascade of contingencies) drastically reducing the maximum power transfer below the pre-contingency demand. Thus, any attempt at restoring power to the pre-contingency demand will induce an unstable response leading to voltage collapse.

As the load dynamics are the driving force of voltage instability, the time scale of load restoration is the one characterizing voltage stability. Thus, fast recovering loads, such as induction motors and power electronics-driven devices, tend to restore load in a second or less and constitute what is known in power system

dynamic analysis as the short-term time scale (Kundur et al. 2004). Study of relevant problems (motor stalling, etc.) is part of *short-term voltage stability* analysis.

In a slower time scale of several seconds up to minutes, load recovery dynamics include the LTCs and thermostatically controlled loads. This is the time scale of *long-term voltage stability* analysis (Kundur et al. 2004). Note that for long-term voltage stability, the short-term dynamics such as those of motors and generators are considered to be in equilibrium. In system representation this assumption leads to the replacement of short-term differential equations with algebraic equilibrium equations. This assumption is known as the quasi-steady-state (QSS) approximation.

Multiple-Load (Meshed) System

The single-load system of Fig. 1 serves well in defining the voltage stability problem and helps visualize its significance through the *PV* curve representation and the maximum loading or critical point C. In actual power systems, however, there are multiple loads defining a multidimensional space where it is sometimes tricky to apply the simple concepts of Fig. 1. For instance, it is important to distinguish between the supply system which can be represented by a Thevenin equivalent and the consumption part where loads affect each other and cannot be examined individually, one at a time.

Consider the power system of Fig. 3, where multiple generators are feeding a number of loads through a meshed network represented by the complex admittance matrix **Y**. The steady-state conditions of the system including generation and load are traditionally represented by the power flow equations:

$$(P_{Gi} + \mathrm{j}Q_{Gi}) - (P_{Ai} + \mathrm{j}Q_{Ai}) - \hat{V}_i$$
$$\sum_{j=1}^{N} \hat{V}_i Y_{ij}^* \hat{V}_j^* = 0 \qquad i = 1, \dots, N \quad (5)$$

Using real variables, (5) can be written as

$$\mathbf{g}(\mathbf{x}, \mathbf{p}) = 0 \qquad (6)$$

Power System Voltage Stability, Fig. 3 Meshed power system
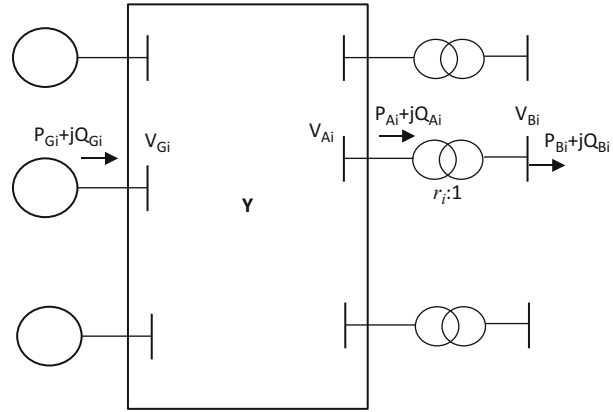
where **p** is the vector of independent parameters (load demands, generator setpoints) and **x** the vector of dependent variables (voltages and angles).

Note that in this representation, the load is referred to the primary side of LTC transformers as in Fig. 1. This can be considered constant at equilibrium corresponding, for instance, to secondary (distribution side) voltage restoration at its setpoint value $V_{Bi} = V_{oi}$.

Concerning generators the active power P_{Gi} cannot be treated as constant when load is varying, so there has to be a participation factor attached to each generator bus that will represent primary or secondary frequency regulation characteristic (Van Cutsem and Vournas 1998). This is sometimes referred to as the *distributed slack bus* approach. For reactive power the limits $Q_{Gi}^{\max}$ of reactive support should be set, beyond which the generator voltage is no longer constant (switch from *PV* to *PQ* bus).

The solution of the N nonlinear complex equations (5) for a given load demand determines all complex voltages in the system. As in the simple radial system case, there may exist multiple solutions, some of which unstable, or no solutions at all. The stability limits, where (3) and (4) hold as equalities for the radial system, are now given by the singularity of the Jacobian of the equilibrium conditions (6):

$$\det \mathbf{D_x g} = 0 \qquad (7)$$

The stability limit can be determined also by the singularity of the state matrix (Medanic et al. 1987; Van Cutsem and Vournas 1998):

$$\det \mathbf{A} = \det \left[\frac{\partial V_i}{\partial r_j} \right] = 0 \qquad (8)$$

Note that the impedance matching condition for a single load amounts to the diagonal element $a_{ii} = 0$ which is much more strict than the singularity condition (8) that marks the actual onset of instability. The points satisfying (7) and (8) are critical points and form a multidimensional manifold in parameter space called *bifurcation surface*.

Applications

The above analysis briefly touches on fundamentals. Detailed analysis tools for voltage stability include (but are not limited to) continuation power flow, VQ curves, time simulation (short-term, long-term, QSS), sensitivity, and eigenvalue/singular value analysis. Voltage security analysis is presently applied online in various control centers based on the above methods of analysis for a large number of contingencies. Countermeasures to voltage instability and collapse cover a wide spectrum, from automatic reactive devices switching to special protection controls and load shedding as a last resort. Further details can be sought in textbooks Taylor (1994) and Van Cutsem and Vournas (1998).

Cross-References

▶ Modeling of Dynamic Systems from First Principles
▶ Stability Theory for Hybrid Dynamical Systems
▶ Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency

Bibliography

Kundur P et al (2004) Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions. IEEE Trans Power Syst 19:1387–1401

Medanic J, Ilic-Spong M, Christensen J (1987) Discrete models of slow voltage dynamics for under load tap-changing transformer coordination. IEEE Trans Power Syst 2:873–882

Taylor CW (1994) Power system voltage stability. EPRI power system engineering series. McGraw-Hill, New York

Van Cutsem T, Vournas C (1998) Voltage stability of electric power systems. Kluwer Academic, Boston (Springer, 2008)

Powertrain Control for Hybrid-Electric and Electric Vehicles

Giorgio Rizzoni
Department of Mechanical and Aerospace Engineering, Center for Automotive Research, The Ohio State University, Columbus, OH, USA

Abstract

Powertrain electrification and hybridization have rapidly become part of the portfolio of all major automotive manufacturers, ranging from hybrid-electric, to plug-in hybrid-electric, to battery-electric vehicles, to hybrid-hydraulic and hybrid-mechanical solutions. The increased complexity of the powertrain systems associated with hybrid vehicles presents interesting control challenges and problems, and this entry describes the more common architectures of hybrid-electric

vehicle powertrains and their operation, focusing on the important problem of optimal control for energy management of hybrid-electric vehicles, on mode switching, and on battery management. In the conclusion, a connection is made between these problems and their interaction with intelligent transportation systems.

Keywords

Introduction

Increasingly stringent fuel economy and emissions regulations have required the automotive industry to consider more fuel-efficient powertrains and alternative primary sources of transportation fuels. Powertrain electrification and hybridization have rapidly become part of the portfolio of all major automotive manufacturers, ranging from hybrid-electric, to plug-in hybrid-electric, to battery-electric vehicles, to hybrid-hydraulic and hybrid-mechanical solutions. The increased complexity of the powertrain systems associated with hybrid vehicles presents interesting control challenges and problems. This entry describes control problems associated with hybrid-electric vehicles (HEVs) and battery-electric vehicles (BEVs).

HEV Powertrains

An HEV powertrain contains at least two power sources: a primary engine – typically a combustion engine or a fuel cell fueled by a chemical fuel (in liquid or gaseous form) – and a secondary power source that makes use of a rechargeable energy storage system (RESS) that permits buffering the power demand of the vehicle so as to provide choices in the use of the power sources. While it is possible to design hybrid powertrains using secondary hydraulic or mechanical energy conversion and storage devices (hydraulic pump/motors and accumulators, mechanical flywheels), the majority of hybrid
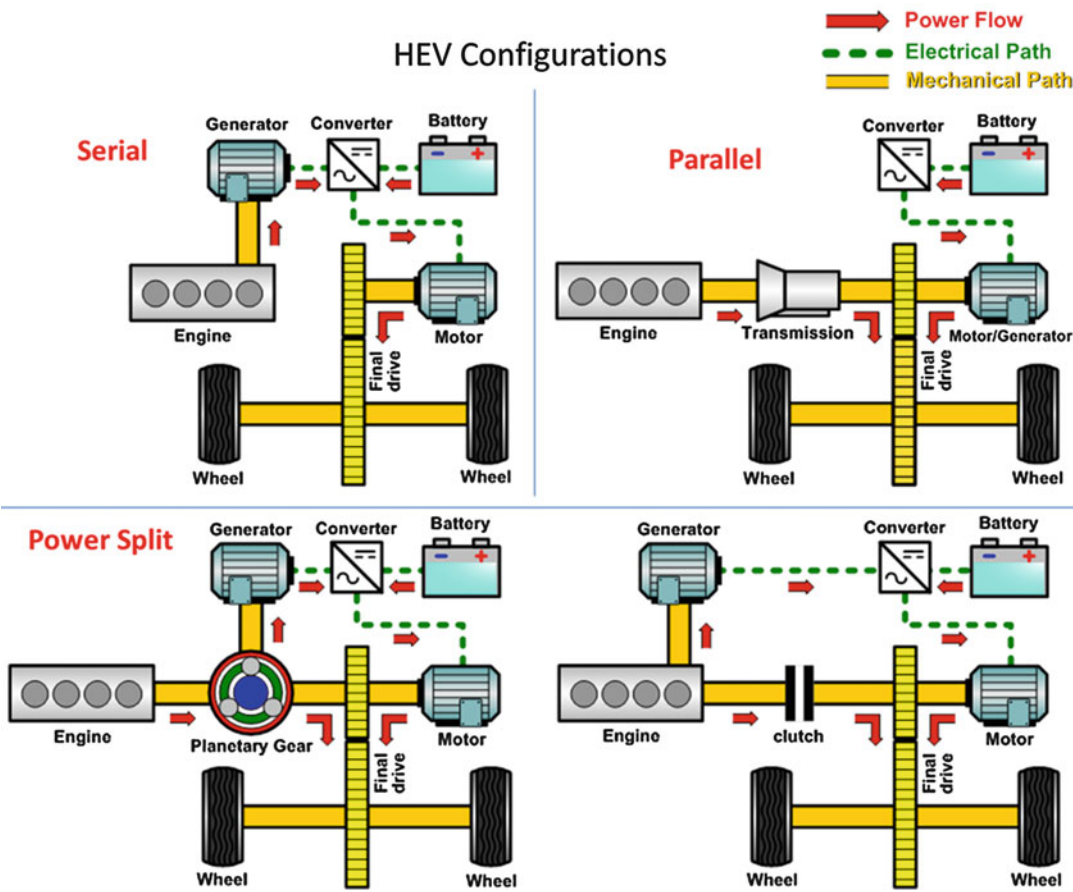
Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 1 Hybrid powertrain configurations (After Rizzoni and Peng (2013), courtesy: Dr. Chiao-Ting Li, the University of Michigan)

powertrains in use today employ electric machines and electrochemical energy storage devices (batteries and supercapacitors); thus, this entry focuses exclusively on *hybrid-electric* vehicles (HEVs). Electric vehicles (EVs) can be viewed as a special case of HEVs in which no internal combustion engine is present, and many of the considerations that follow apply also to hydraulic and mechanical hybrids. HEVs may be classified according to their powertrain architecture as shown in Fig. 1.

A *series HEV powertrain* employs an electric machine (EM) to propel the vehicle while using an internal combustion engine (ICE) coupled to a second EM as an electrical generator set. In a series HEV, the electrical generator set can provide power directly to the electric traction

system, via an electrical DC bus, or can charge an RESS (e.g., battery), or can perform both functions Motive power to the vehicle is delivered by the primary EM. Thus, a series HEV blends electrical power from an RESS with electrical power generated by an ICE-powered generator set to provide motive power to the vehicle. Deciding how much electrical power to draw from each of the two power sources to meet the power demand of the vehicle is an important control objective. A further feature of interest is the ability to recover some of the kinetic energy of the vehicle during braking events by using the traction EM in generator mode to recharge the RESS.

A *parallel HEV powertrain* blends mechanical power from the ICE and one or more EMs

P

through appropriate mechanical coupling and transmission elements to deliver motive power to the vehicle or to recharge the RESS. In a parallel HEV powertrain, the same EM is used to provide power to the vehicle (motor mode) and to provide energy to the RESS (generator mode); in the latter case, the RESS can be recharged either by providing power from the ICE in excess of that required by the vehicle or by converting the kinetic energy of the vehicle into electrical power through the braking action of the EM.

A third configuration, the one that is most commonly found among passenger vehicles in commercial production today, is the *power-split HEV*, in which the benefits of both series and parallel HEVs are achieved most commonly by using one or more planetary gear sets to couple two EMs – to the ICE on one side and to the driveline on the other.

Regardless of architecture, HEV powertrains enable fuel savings and emissions reductions by operating in a variety of modes that include *load leveling*, *regenerative braking*, *engine start-stop*, and *transmission optimization* (Miller 2004; Rizzoni and Peng 2013). All of these functions benefit from the availability of an RESS and of bidirectional power converters, that is, the electric drive system(s) that can serve both motor and generator functions.

HEV Operation

An HEV is considered *charge sustaining* if the RESS is recharged only by power supplied by the ICE or by regenerative braking. If, on the other hand, the vehicle is designed to deplete energy stored in the RESS during the course of a trip, ending the trip with a lower state of stored energy than at the start and requiring recharging from the electrical grid, the vehicle is called *charge depleting* and is commonly referred to as a *plug-in HEV* (PHEV). PHEVs can in turn be subdivided into blended-mode PHEVs, in which stored electrical energy and fuel chemical energy are used jointly to achieve minimum overall energy use, and extended-range electric vehicles (EREVs), in which electrical energy is used exclusively to power the vehicle, until a lower bound is reached,

at which point the vehicle uses both ICE and EM(s) to behave like a charge-sustaining hybrid. In principle, any of the architectures of Fig. 1 can be used in any of these modes. A battery-electric vehicle, or BEV, is an extreme case of an EREV, in which the vehicle is not equipped with an ICE. Miller (2004) provides an excellent overview of the technology underlying each of the powertrain architectures mentioned so far.

Control Problems in X-EVs

Let us refer to the general case of a hybrid or electric vehicle as an X-EV, with the X in X-EV representing any of the architecture discussed so far: X− = H, PH, ER, or B. X-EVs enable multiple configurations and operating modes of the powertrain, presenting a number of interesting control problems above and beyond those that are already present in non-hybrid powertrains (e.g., engine and transmission control). In general, the control architecture of an HEV is hierarchical, with a higher-level (supervisory) controller that manages the power flows and mode changes (e.g., from electric to hybrid in an EREV) to meet the vehicle fuel economy, emissions, performance, and drivability requirements. Figure 2 depicts a hierarchical control architecture in use in a prototype PHEV.

In an X-EV, two problems are especially important: *optimal energy management*, that is, the ability to optimize the energy use of a vehicle during a trip, and *mode switching*, that is, the ability to select the appropriate operating mode and to smoothly switch between modes.

The higher-level controller issues set points to lower-level controllers that are used to manage the ICE, the EM(s), the mechanical transmission system, the brake system, and the RESS, as well as other auxiliary functions in the vehicle. In this article we primarily consider the higher-level controller and focus on two problems that are especially relevant to HEVs: *optimal energy management* and *mode switching*. In addition, we also consider the battery controller (often called *battery*
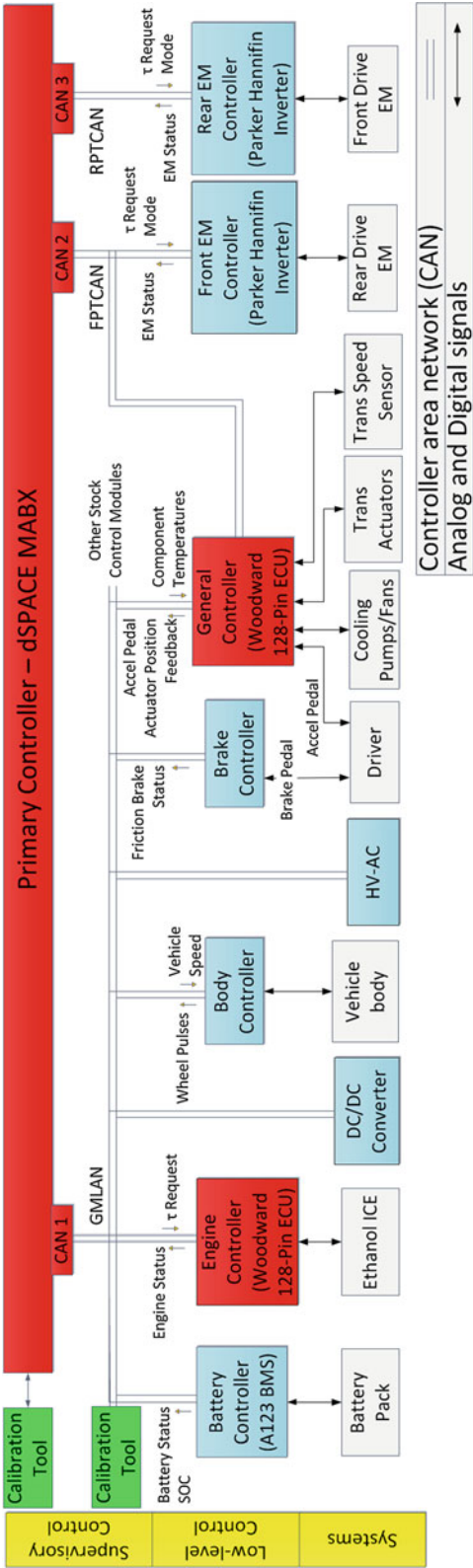
Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 2 Hierarchical structure of an HEV controller (Courtesy: The Ohio State University EcoCAR 2 Team)

P

management system or BMS), which while being a low-level control is very specific to X-EVs.

Optimal Energy Management

The optimal energy management problem in an X-EV consists of finding the control $u(t)$ that leads to the minimization of a performance index J over the time horizon $t - t_f$, corresponding to a driving cycle, or *trip*; the problem is subject to constraints that are related:

(i) To physical limitations of the actuators and the energy stored in the RESS
(ii) To the requirement to maintain the RESS state of energy within prescribed limits (in a charge-sustaining X-EV) or to track a specified RESS stored energy trajectory (in charge-depleting X-EVs)

Let $L(\cdot)$ be a suitable function of the system states and inputs that accounts for the quantities we wish to minimize, for example, fuel consumption or emissions of carbon dioxide. Then, we define the cost function

$$J(x(t), u(t)) = \int_{t_0}^{t_f} L(x(t), u(t), t) dt \quad (1)$$

which is to be minimized for every trip. In general, the exact driving cycle, or profile, associated with a trip is not completely known; thus, a causal solution to this problem is impossible to achieve without making some assumptions. Various approaches to solve (1) have been proposed over the years; we cite (i) dynamic programming (DP), (ii) local optimization solutions as surrogates of a global solution, (iii) Pontryagin's minimum principle (PMP), and (iv) rule-based methods. Onori et al. (2014) provide a comprehensive overview of the problem as well as detailed examples. We briefly review approaches (i), (ii), and (iii) in the present article.

Global Optimization by Dynamic Programming

If the driving cycle, represented by the vehicle instantaneous velocity over time, $v(t)$ is known, it is possible to cast (1) in such a form that

a DP solution is possible. For example, in a charge-sustaining X-EV, one can find the sequence of inputs that minimizes the trip fuel consumption while sustaining the desired state of charge of a battery and meeting the speed profile of the vehicle. In this problem, the input is the power supplied by the battery to the electric machine, and the state of charge of the battery, SOC, is the only state; all other subsystems (engine, electric drives, transmission, etc.) are modeled via quasi-static efficiency models that can be represented by algebraic equations (e.g., Willans lines, Rizzoni et al. 1999) or by maps. The vehicle velocity profile, $v(t)$ is converted to a vehicle power request, $P_{REQ}(t)$, knowing the vehicle load characteristics (aerodynamic, inertial, rolling and drivetrain friction, and road grade). In turn, the power required to meet a specific load profile is the sum of the power delivered by the ICE and EM, $P_{REQ}(t) = P_{ICE}(t) + P_{BAT}(t)$. So, for example, we seek the control input, $P_{BAT}(t)$, that corresponds to the minimum fuel consumption, that is, $\min\limits_{\{P_{ICE}(t), P_{BAT}(t) \forall t\}} \int_{t_0}^{t_f} \dot{m}_f(t) dt$, while delivering the requested vehicle power. The problem has physical constraints in the actuators (maximum and minimum power that can be delivered by ICE and EM), as well as the requirement for the control policy to be charge sustaining, which is translated into the additional condition $SOC(t_0) = SOC(t_f)$. While this is only a sketch of the problem formulation (see Onori et al. (2014) for a detailed treatment), it should be clear that it is possible to find a DP solution. If the vehicle is charge depleting, the problem can be similarly formulated with $SOC(t_f) < SOC(t_0)$.

In practice, this approach requires complete information of the vehicle velocity profile, and DP is not an implementable, causal solution to the X-EV energy management problem. It is, however, a very useful tool to establish a benchmark for a problem or as an aid in developing a rule base (Onori et al. 2014). Stochastic DP methods have been proposed to circumvent the need to know the driving cycle exactly (see, e.g., Tate et al. 2007).

Local Optimization by Equivalent Fuel Consumption Minimization

A heuristic approach that has met with success is to solve (1) as a local optimization problem, wherein $\int_{t_0}^{t_f} \min_{\{P_{ICE}(t), P_{BAT}(t) \, \forall t\}} \dot{m}_f(t) dt$ is used as an approximation for $\min_{\{P_{ICE}(t), P_{BAT}(t) \, \forall t\}} \int_{t_0}^{t_f} \dot{m}_f(t) dt$. This approach gives rise to the *Equivalent fuel Consumption Minimization Strategy* (ECMS) (Paganelli et al. 2001), which accounts for the use of stored electrical energy, in units of chemical fuel use (g/s), such that one can define an "equivalent fuel consumption" taking into account the cost of the electrical energy used to produce $P_{BAT}(t)$ by way of the fuel that must be used at a future time to replenish the stored electrical energy in the RESS. The equivalent fuel consumption is defined in (2):

$$\dot{m}_{f,eq}(t) = \dot{m}_f(t) + \dot{m}_{eq}(t) = \dot{m}_f(t)$$
$$+ s(t) \frac{E_{batt}}{Q_{LHV}} S\dot{O}C(t) \quad (2)$$

In (2), $\dot{m}_{f,eq}$ is the equivalent fuel consumption, $\dot{m}_f$ is the actual chemical fuel consumption, $\dot{m}_{eq}$ is the virtual fuel consumption corresponding to the use of electricity stored in the battery (to be replenished in the future), E_{BAT} is the energy capacity of the battery, Q_{LHV} is the lower heating value of the chemical fuel, and $s(t)$ is the *equivalence factor* that assigns a cost to the use of electricity. Then, the global minimization problem of (1), with $L(\cdot)$ equal to $\dot{m}_{f,eq}$, becomes the problem of finding $\int_{t_0}^{t_f} \min_{\{P_{ICE}(t), P_{BAT}(t) \, \forall t\}} \dot{m}_f(t) dt$. This approach, which can be easily implemented, has been used widely and has been shown to closely approximate the global optimal solution if sufficient knowledge of the vehicle driving cycle is available. The method does requires empirical calibration and tuning of the equivalence factor, $s(t)$, the optimal value of which is dependent on the driving cycle. Such calibration could be automated by using a predictor to generate a short-horizon estimate of the driving cycle and an adaptor to generate an appropriate $s(t)$ (Musardo et al. 2005).

Optimization by Pontryagin's Minimum Principle

Pontryagin's minimum principle (PMP) can also be employed to solve the X-EV energy management problem. If, again, the fast dynamics of the system are neglected the state equation is

$$\dot{x}(t) = f(x, u, t) = -\frac{1}{E_{BAT}} I_{BAT}(x, u, t) \quad (3)$$

where $x = SOC$ is the state of charge of the battery, E_{BAT} is the energy capacity of the battery, and I_{BAT} is the instantaneous battery current. If the input is the power requested of the battery, $P_{BAT}(t)$, which in turn determines the engine power request, $P_{ICE}(t)$, and hence the fuel consumption, then the Hamiltonian function can be defined to be

$$H(x(t), P_{BAT}(t), \lambda(t)) = \dot{m}_f(P_{BAT}(t)) - \lambda(t) \cdot$$
$$f(x(t), P_{BAT}(t), t) \, (4)$$

In (4), $f(\cdot)$ is given by Eq. (3), and the control $P_{BAT}(t)$ that which minimizes Eq. (4) at each time instant is
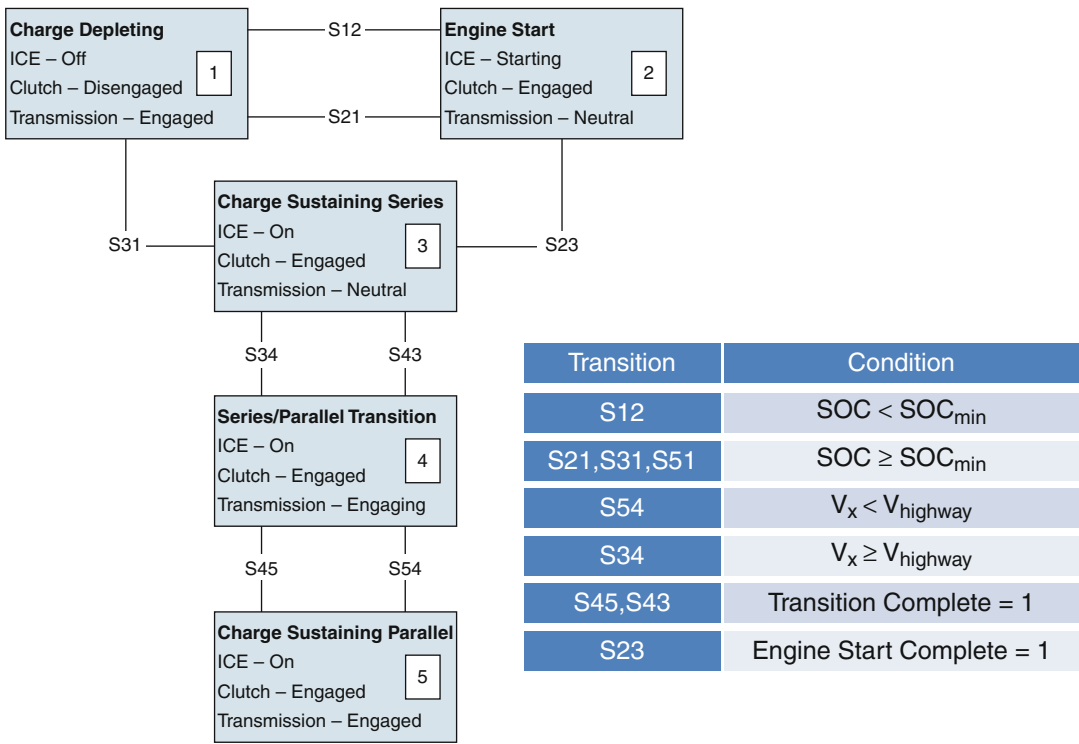
$$P_{BAT}^*(t) = \arg\min_{P_{BAT}} H(x(t), P_{BAT}(t), \lambda(t)) \quad (5)$$

The co-state variable, $\lambda(t)$, is the solution of

$$\dot{\lambda}(t) = -\lambda(t) \frac{\partial f(x(t), u(t))}{\partial x} \quad (6)$$

Eqs. 3 and 5, with boundary conditions $x(t_0)$ and $x(t_f)$, can be solved numerically; in Serrao et al. (2009, 2011) it is shown that the co-state $\lambda(t)$ is related to the *equivalence factor* of Eq. (2), confirming that the intuitive ECMS solution is in fact the PMP solution, providing that the equivalence factor (or co-state) is time varying and satisfies

$$H(t, x, u, \lambda) = \dot{m}_f + \lambda(t)\dot{x}(t) \quad \text{and}$$
$$s(t) = -\lambda(t) \frac{Q_{LHV}}{E_{BAT}} \quad (7)$$

Charge Depleting		Engine Start	
ICE – Off		ICE – Starting	
Clutch – Disengaged	1	Clutch – Engaged	2
Transmission – Engaged		Transmission – Neutral	

Charge Sustaining Series	
ICE – On	
Clutch – Engaged	3
Transmission – Neutral	

Series/Parallel Transition	
ICE – On	
Clutch – Engaged	4
Transmission – Engaging	

Charge Sustaining Parallel	
ICE – On	
Clutch – Engaged	5
Transmission – Engaged	

Transition	Condition
S12	$SOC < SOC_{min}$
S21,S31,S51	$SOC \geq SOC_{min}$
S54	$V_x < V_{highway}$
S34	$V_x \geq V_{highway}$
S45,S43	Transition Complete = 1
S23	Engine Start Complete = 1

Powertrain Control for Hybrid-Electric and Electric Vehicles, Fig. 3 State diagram illustrating mode switching in a PHEV (Courtesy: The Ohio State University EcoCAR 2 Team)

The PMP solution is also cycle dependent, as the optimal initial condition for the co-state is dependent on the driving cycle. This dependence on the driving cycle, whether expressed in terms of an equivalent fuel consumption in the ECMS solution or as the initial condition of the co-state in the PMP solution, is an unavoidable consequence of the fact that the fuel consumption of a vehicle is strongly dependent on the driving conditions, which affect the vehicle load.

The basic concepts outlined above continue to be the subject of further development; for example, integrating available trip information available from navigation and geographical information systems into predictive energy management algorithms and considering battery aging as a cost in the optimization function are but two of the research areas being pursued.

Mode Switching

X-EV architectures permit multiple operating modes to exploit the design and control flexibility available in the powertrain. Some examples are the following: an X-EV could operate in pure EV mode or in hybrid mode (whether series, parallel, or power-split), could use special control algorithms during regenerative braking events to provide maximum energy recovery without adversely affecting brake and vehicle stability control systems, and could implement special start-stop control strategies that minimize fuel consumption at idle without adversely affecting engine cold- or warm-start emissions and without inducing unwanted transient vibrations (Canova et al. 2009). Figure 3 depicts an example of a state flow diagram that could be implemented in a finite state machine. Mode switching can result in *drivability* problems (Wei and Rizzoni 2004), that is, in undesirable

transient response characteristics during mode changes. An X-EV can, in this context, be represented as a hybrid system (Koprubasi et al. 2007).

Battery Management Systems

The most common RESS in hybrid vehicle is the electrochemical battery. A hybrid or electric vehicle uses a battery pack that is typically composed of modules, which are in turn comprised of battery cells connected in series and parallel. Battery management systems are necessary to provide charge balancing, cell protection, state of charge and state of health estimation, and other functions related to the management of the stored energy. A good overview of battery systems and associated control problems may be found in Rahn and Wang (2013).

Two important problems related to battery management are state of charge (SOC) and state of health (SOH) estimation. SOC estimation is a necessary component of any battery management system. The SOC of battery is defined by the following equations, in which x is the SOC, Q_{BAT} is the battery capacity in ampere-hours, and η is the battery charging/discharging efficiency:

$$\dot{x}(t) = \frac{\eta}{Q_{BAT}(t)} \cdot I_{BAT}(t) \quad x(t) = x(t_0)$$

$$+ \frac{1}{3,600 \cdot Q_{BAT}(t)} \int_{t_0}^{t_f} I_{BAT}(\tau) \cdot d\tau \quad (8)$$

In practice, there are two problems with using current integration (also called *Coulomb counting*) to estimating SOC: (i) errors in numerical integration accumulate and may cause significant bias error in the estimate, and (ii) the actual capacity of the battery is unknown during vehicle operation, as it changes over time due to battery aging. A second SOC estimation approach consists of correlating the battery open-circuit voltage to the SOC, but this approach also suffers from significant uncertainty, as the open-circuit voltage-SOC correlation curves are only accurate in stationary conditions (constant temperature, with battery at rest). SOC estimation has been the subject of

much research and has seen the use of Kalman filters, extended Kalman filters, particle filters, and other estimation approaches (Chaturvedi et al. 2010).

The SOH of a battery degrades over time due to two principal factors: *capacity fade* and *power fade* (which can also be thought of as *conductance fade* caused by an increase in the internal resistance of the battery). These phenomena are the result of complex electrochemical interactions that are specific to battery chemistry. The ability to estimate the capacity and resistance of a battery during actual operation is a very important aspect of battery management. As in the case of SOC estimation, no direct measurement is possible outside of controlled laboratory conditions; hence, estimation algorithms must be employed (Chaturvedi et al. 2010). It is important to observe that SOC and SOH estimation algorithms operate on two completely different time scales, as the SOC of a battery fluctuates over time windows of minutes or hours, while the SOH changes very slowly over time, with measurable changes occurring over periods of months or years.

Summary and Future Directions

In summary, the control of X-EV powertrains is a rich subject for control theoreticians and practitioners, presenting topics related to optimization and optimal control (for energy management, battery aging), hybrid control (for drivability), adaptive and predictive control, and estimation. Further, the electrification of ground vehicles presents interesting opportunities to integrate vehicles with the electric power and communication networks infrastructures. The following paragraphs describe two such opportunities.

Vehicle-Grid Interaction

As the penetration of plug-in vehicles, PHEVs and BEVs, increases, their impact on the electric power grid cannot be neglected; the consideration of increased electric power demand and of the

timing of vehicle charging must be included in the control/optimization of the electric power grid.

The electric grid and the transportation system are the two largest sectors that produce greenhouse gas emissions. When large numbers of vehicles are electrified and draw power from the electric grid, it is important to aim for reduced overall greenhouse gas emissions rather than just shifting emissions from tailpipes to power plant stacks. Controlling the charging of plug-in vehicles to alleviate the impact to the grid has been studied, including the idea of using plug-in vehicles as ancillary services to the grid, possibly with significant renewable power sources connected to the grid. Modeling and simulating this integrated system require information on detailed grid load profiles, power generation pricing and carbon emissions, wind statistics, and vehicle usage statistics. In addition, charging control must balance multiple factors: grid stability, fully charging all vehicles, minimizing data collection and communication, and overall system carbon emission minimization.

Intelligent Transportation Systems

X-EVs, as well as conventional vehicles, will benefit from the ability to analyze traffic and geographical information in real time to quantify the effects of infrastructure, environment, and traffic flow on vehicle fuel economy and emissions, and to permit the application of forecasting and optimization methods for energy management (Gong et al. 2011; Wollaeger et al. 2012). There are significant opportunities to achieve significant fuel savings and emissions reduction by considering the large-scale interactions of vehicles with one another and with the infrastructure, further exploiting the flexibility inherent in X-EVs.

Cross-References

Bibliography

Canova M, Guezennec Y, Yurkovich S (2009) On the control of engine start/stop dynamics in a hybrid electric vehicle. ASME J Dyn Syst Meas Control 131: 061005

Chaturvedi NA, Klein R, Christensen J, Ahmed J, Kojic A (2010) Algorithms for advanced battery management systems. IEEE Control Syst Mag 30(2):49–68

Gong Q, Tulpule P, Midlam-Mohler S, Marano V, Rizzoni G (2011) The role of ITS in PHEV performance improvement. In: American control conference, San Francisco

Koprubasi K, Westervelt ER, Rizzoni G (2007) Toward the systematic design of controllers for smooth hybrid electric vehicle mode changes. In: Proceedings of the American control conference, Anchorage

Miller JM (2004) Propulsion systems for hybrid vehicles. The Institution of Electrical Engineers, London

Musardo C, Rizzoni G, Guezennec Y, Staccia B (2005) A-ECMS: an adaptive algorithm for hybrid electric vehicle energy management. Eur J Control 11(4–5): 509–524

Onori S, Serrao L, Rizzoni G (2014) Energy management strategies for hybrid electric vehicles. Springer, Berlin

Paganelli G, Ercole G, Brahma A, Guezennec Y, Rizzoni G (2001) General supervisory control policy for the energy optimization of charge-sustaining hybrid electric vehicles. JSAE 22: 511–518

Rahn C, Wang C-Y (2013) Battery systems engineering. Wiley, New York

Rizzoni G, Peng H (2013) Hybrid and electric vehicles: the role of dynamics and control. ASME Dyn Syst Control Mag 1(1):10–17

Rizzoni G, Guzzella L, Baumann B (1999) Unified modeling of hybrid-electric vehicle drivetrains. IEEE/ASME Trans Mechatron 4(3):246–257

Serrao L, Onori S, Rizzoni G (2009) ECMS as a realization of Pontryagin's minimum principle for HEV control. In: Proceedings of the 2009 American control conference, Portland

Serrao L, Onori S, Rizzoni G (2011) A comparative analysis of energy management strategies for hybrid electric vehicles. ASME J Dyn Syst Meas Control 133:1–9

Tate ED, Grizzle JW, Peng H (2007) Shortest path stochastic control for hybrid electric vehicles. Int J Robust Nonlinear Control 18(14):1409–1429

Wei X, Rizzoni G (2004) Objective metrics of fuel economy, performance and driveability – a review. SAE Technical paper 2004-01-1338

Wollaeger SK, Onori S, Di Cairano S, Filev D, Ozguner U, Rizzoni G (2012) Cloud-computing based velocity profile generation for minimum fuel consumption: a dynamic programming based solution. In: American control conference, Montreal, 27–29 June 2012

Programmable Logic Controllers

Georg Frey
Saarland University, Saarbrücken, Germany

Synonyms

PLC

Abstract

Programmable logic controllers (PLCs) are a special form of computing hardware and software tailored for use in industrial control. The hardware is built for rough environments and offers various input and output ports for industrial sensor and actuator signals as well as communication systems. The main software features are hard real-time capabilities and a set of standardized programming languages specifically designed for the realization of automation functions.

Keywords

Function blocks; Ladder diagram; Ladder logic; Logic control; Real-time control

Introduction

Since the 1970s, the programmable logic controller (PLC) has been the primary workhorse of industrial automation. For a long time, it has provided a distinct field of research, development, and application, mainly for control engineering. This area has produced its own design methods and programming languages. Due to its importance for industrial application, a lot of these methods have been standardized by the International Electrotechnical Commission (IEC). Currently the most influential standards are IEC 61131 (John and Tiegelkamp 2010) and IEC 61499 (Vyatkin 2011). While the latter one is dedicated to distributed systems, IEC 61131 covers the PLC as such. This standard consists of several parts. The most important ones are:

Part 1 : General information. This part covers the *CONCEPT* of PLCs. It describes the general idea and typical functionalities, most importantly, the cyclic processing of the application program working on a stored image of the input and output values.

Part 2 : Equipment requirements and tests. Here requirements on the PLC *HARDWARE* (electrical, mechanical, and functional) and corresponding tests are defined.

Part 3 : Programming languages. This is the most important part of the standard. Based on already existing PLC programming languages, a harmonization of the *SOFTWARE* structure was achieved. This includes a general software model together with a set of different standardized programming languages. IEC 61131-3 paved the way from proprietary programming solutions to a set of well-accepted languages, allowing easier training of PLC programmers and – to some extent – the reuse of application solutions on different hardware platforms.

While Part 2 is of importance for PLC manufacturers only, Parts 1 and 3 contain relevant information for PLC users, especially for designers of PLC control applications. Before discussing these points, the definition of PLC from IEC 61131-1 is reproduced and discussed:

> A PLC is a digital electrical system used in manufacturing. It utilizes programmable memory to store practice-oriented control programs. Thus is suitable for implementation of specific functions such as combinatorial control, sequence control, time-, count- and arithmetic functions. Due to its special arrangement of digital or analog input/output, it is used for controlling various machines and processes. (. . .)

This definition is focused on the usage of the device and would – taken out of the context – also cover industrial PCs or microcontroller-based control solutions. The specifics of PLC hardware

P

are discussed in Part 2 of the standard. However, much more important for distinguishing a PLC from other control hardware are the properties of the execution model described in Part 2 and discussed in the following.

Execution Model

In designing PLC applications, the execution model has to be considered. The main idea is the cyclic execution together with an I/O image. While microcontrollers and PCs typically use an event-based execution model (the application waits for external events from the environment – interrupts – and reacts accordingly), the PLC follows a time-based scheme (the application scans the environment at instances in time – often a fixed cycle time – and reacts on the new status of the input ports).
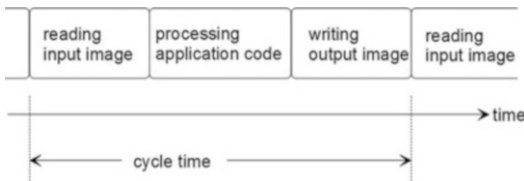
A PLC cycle consists of three iterated steps: input reading, program execution, and output writing. Together with the concept of the process image – a reserved memory space where input and output variables are stored – this execution model leads to the following:

(a) During one cycle, input and output values are kept fixed, i.e., a change in input signal values during a cycle will not be seen by the program executed. This means that a temporal change in an input signal value that is shorter than the cycle time may not be registered by the PLC at all.

(b) Changes in output signal settings by the program will be switched to the actual output ports only after execution of the complete program. This actually means that for an output signal where the value is changed several times during one program execution, only the last change will be set to the hardware output of the PLC.

(c) The response time of a PLC, i.e., the time between a change in an input signal and the corresponding reaction at the output port of a PLC, lies between one and two PLC cycles, depending on when the change at the input port occurs relative to the PLC cycle.

While the time needed for input reading and output writing is constant over all cycles, the time for program execution may vary due to conditional execution of some program parts. However, normally the PLC is operated with a fixed cycle time set high enough to allow for the worst-case execution time of the application program.

The advantage of the described concept is the deterministic behavior of the resulting system with a very simple way to determine the timing behavior. This is important for most PLC applications:

(a) Open-loop control, where the reaction to a change of an input signal has to be reached in a limited time, especially in safety-critical applications.

(b) Closed-loop control, where the design of a discrete-time control algorithm is based on the assumption of a fixed sample-time.

To realize control functions, an application program has to be written for the PLC. To this end, Part 3 of the IEC 61131 defines a software model together with a set of programming languages.

Software Model and Programming

The original idea that led to the development of the first programmable logic controller (PLC) in 1968 was to replace hardwired control equipment at machines. Back then, the controllers of machines, for example, lathes or grinders, typically consisted of a cabinet of interconnected relays. The size of such a controller could be considerable and its failure rate was high due to mechanical defects of single relays. Furthermore, the initial setup was very time-consuming and error prone, because the relays (often hundreds of them) had to be wired by hand. The biggest

drawback of this technology, however, was the problems arising if a controller had to be changed, employing a new function or adjusting to a new production task. Then the hardwired structure had at least partially to be disassembled and rewired. Here was the main advantage of a controller that could be adjusted by changing software instead of hardware.

Since the first PLCs in the early seventies reached the market, graphical programming methods are used to develop the control algorithms. These are ladder diagram (LD, sometimes also referred to as ladder logic) and later function block diagram (FBD). The implementation of LD on the very first PLC (the Modicon 084) was intended to allow an easy access for the people doing hardwired relay logic until then. (More on the history of PLCs can be found on the website of Dick Morley, commonly known as the father of the PLC (http://www.barn.org/FILES/historyofplc.html).)

LD, at least in its early forms, is basically the graphical representation of its hardwired forefather. The name ladder comes from the fact that on both sides of the drawing, there is a power rail and horizontally between those rails, like rungs on a ladder, sequences of logical element are drawn. The basic of these elements are relays (switches), depending on input signals or internal variables, and coils (memories to store variables and set output signals). The ladder is processed in a top-down and left-right fashion.

Figure 1 shows an example of an LD. Every rung can be read as an IF THEN ELSE statement. The first rung of the ladder means IF (Var1 $= 1$ AND Var2 $= 1$) THEN (Var3 $:= 1$; Var4 $:= 0$) ELSE (Var3 $:= 0$; Var4 $:= 1$). The second rung is IF (Var3 $= 1$ OR Var4 $= 0$) THEN (Var1 $:= 1$) ELSE (Var1 $:= 0$).

While LD resembles relay logic, FBD is a graphical mimicking of the wiring of simple logic gates, like AND, OR, NOT, or FLIP-FLOP. Both languages (LD as well as FBD) are still part of the IEC 61131-3. However, they are not well suited for the description of sequential and concurrent algorithms because they have no means
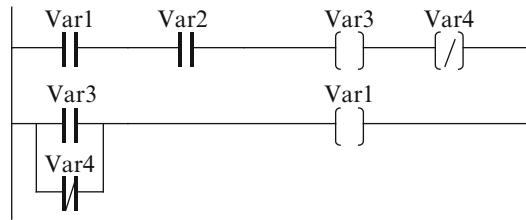
Programmable Logic Controllers, Fig. 1 Example of a PLC program written in Ladder Diagram (LD)

for the visual description of the control flow in a program.

The IEC 61131-3 standard also contains a language that is intended for the graphical description of sequential and concurrent behavior: the sequential function chart (SFC). The SFC is based on Grafcet (David 1995) and represents a form of Petri net (with very special dynamics and functionality). Due to its high functionality, SFC can be easily applied for the structuring of a PLC program on a high level. However, it is cumbersome (and by the standard also not intended) to use for the specification of a low-level sequential algorithm, as, for example, the alternative switching between two motors.

In addition to the three graphical languages, there are also two textual languages in the standard: the assembler-like Instruction List (IL) and the Pascal-like Structured Text (ST).

The decision for one of the languages is based on functional aspects of the application to be realized (high-level languages SFC and ST vs. low-level languages LD, IL, and FBD) but also on traditions in the application domain (e.g., LD in automotive manufacturing vs. FBD in process industry), the geographical region (e.g., LD in the US vs. IL in Germany), and the preferences of the programmer (graphical vs. textual). To allow for flexible solutions and the optimal choice of languages, IEC 61131-3 allows the use of different languages for different parts of the control application.

An application in IEC 61131-3 is structured into program organization units (POUs). Each of the POUs contains a header in a unified syntax for parameter and variable definitions and a body for

P

the actual program code. This body can be written in any one of the defined PLC languages.

There are three types of POUs: Program, Function Block, and Function. A program is the top-level POU of a PLC application. Only in a program, variables can be linked to actual input and output ports. A program can call Function Blocks which in turn may call other function blocks. Programs and function blocks can also call Functions. A POU of type function has no internal memory while a Function Block has memory.

IEC 61131-3 introduced the type and instance concept into PLC programming. A Function Block is always the instantiation of a Function Block Type. Each instantiation gets its own name and variable space. This concept is similar to – but much older than – the class-object instantiation idea of object-oriented programming languages. The exclusive use of symbolic variables without direct references to hardware addresses or ports in Function Blocks allows their easy reuse in one or more applications and the definition of widely applicable Function Block (Type) Libraries.

Summary and Future Directions

PLCs are a proven technology in industrial automation. They follow a simple but deterministic execution and software model. This is the main reason why PLCs are still here and will be here for quite some time to come even if there is faster and fancier technology like embedded PCs available.

Currently the third edition of IEC 61131-3 is nearly ready for publishing. In addition to minor corrections, this new edition adds some concepts from object-oriented programming to the existing software model. First tools on the market already support these extensions.

For the future, two trends can be seen. First, there is a growing trend to integrate PLC programming into model-based software development processes: either by generating PLC code from existing model-based toolchains or by integrating model-based approaches, especially from

the object-oriented domain, into PLC programming environments. Either way this is due to the fact that the complexity in PLC application is rising while the development time should be decreased.

Second, there is a growing interest in the use of formal methods in the PLC domain. In recent years, a lot of interdisciplinary work was aimed in this direction. This work results in the formalization of different steps in the control design process depending on what problems are to be solved (Frey and Litz 2000):

1. The demand for reduced development time and the possible reuse of existing software modules result in the need for a formal approach to the development of the PLC programs.
2. The demand for high-quality solutions and especially the application of PLC in safety-critical processes result in the need for validation procedures, i.e., formal methods to prove specific static and dynamic properties of the programs.
3. The large numbers of already installed PLC programs, together with the high expense of programming, lead to the search for verification and validation methods that can be applied directly to programs written in PLC-specific programming languages such as ladder diagram.

To conclude, more than 50 years after its invention, the PLC is still an industrial success story, and due to ever-increasing demands on the complexity and correctness of its applications, it also still provides much room for further research and development.

Cross-References

Bibliography

David R (1995) Grafcet: a powerful tool for specification of logic controllers. IEEE Trans Control Syst Technol 3:253–268

Frey G, Litz L (2000) Formal methods in PLC programming. In: Proceedings of the IEEE conference on systems man and cybernetics SMC 2000, Nashville, Tennessee, pp 2431–2436

John K, Tiegelkamp M (2010) IEC 61131-3: programming industrial automation system: concepts and programming languages, requirements for programming systems, aids to decision-making tools, 2nd edn. Springer, Berlin

Vyatkin V (2011) IEC 61499 as enabler of distributed and intelligent automation: state-of-the-art review. IEEE Trans Ind Inform 7:768–781

Proportional-Integral-Derivative Control

▶ PID Control

Pursuit-Evasion Games and Zero-Sum Two-Person Differential Games

Pierre Bernhard
INRIA-Sophia Antipolis Méditerranée, Sophia Antipolis, France

Abstract

Differential games arose from the investigation, by Rufus Isaacs in the 1950s, of pursuit-evasion problems. In these problems, *closed-loop strategies* are of the essence, although defining what is exactly meant by this phrase, and what is the "Value" of a differential game, is difficult. For closed-loop strategies, there is no such thing as a "two-sided maximum principle," and one must resort to the analysis of Isaacs' equation, a Hamilton Jacobi equation. The concept of viscosity solutions of Hamilton-Jacobi equations has helped solve several of these issues.

Keywords

Historical Perspective

The history of differential games (DG in short) starts with Rufus Isaacs, who coined the phrase in his pioneering work of the early 1950s (Isaacs 1951), which was largely ignored until the publication of his book Isaacs (1965). Through the investigation of particular problems, Isaacs invented by himself (with his own names) the concepts of state and control variables, of feedback, his "tenet of transition" – better known as Bellman's optimality principle – the (Hamilton-Jacobi-Caratheodory-)Isaacs equation, barriers, some difficult corner conditions ("equivocal lines"), singular arcs ("universal lines"), etc.

Another very early work was Kelendzerize's chapter "A Pursuit Problem" in the historical book by Pontryagin et al. (1962), but it lacked closed-loop strategies.

John Breakwell and a few followers (Breakwell and Merz 1969; Breakwell 1977) picked up Isaacs' work where he had left it, still working on particular problems, but adding the power of the computer to analyze the solution of Isaacs' equation via the structure and singularities of fields of extremal trajectories, while most of the literature concentrated on making precise the concepts of closed-loop strategies and of the Value of the game. Prominent figures in that quest are Krasovskii and Subbotin (1977), Fleming (1961), Friedman (1971), Blaquière et al. (1969), Elliot and Kalton (1972), Emilio and Roxin (1969), and Varaiya and Lin (1969) who together invented the concept of non-anticipative strategies.

The major later innovation was Crandall and Lions' viscosity solutions of PDEs (Crandal and Lions 1983; Lions 1982) applied to DGs and its Isaacs equation by Evans and Souganidis (1984) and Lions and Souganidis (1985).

We also refer the reader to the entry (Quincampoix 2009) of another Springer Encyclopedia.

General Setup

We shall be interested in (continuous time) two-person zero-sum DGs with complete information, this last phrase meaning that both players know exactly and instantly the state of the system, but (usually) not their opponent's control.

The available space of a short article does not allow us to attempt to give the most general setup of a zero-sum two-person perfect-information differential game. We shall therefore concentrate on a typical class, with a finite dimensional state space, as follows. The data are:

1. A two-player dynamical system with state $x \in \mathbb{R}^n$, control variables $u \in \mathsf{U} \subset \mathbb{R}^\ell$, $v \in \mathsf{V} \subset \mathbb{R}^m$ (U and V will often be assumed compact), and its dynamics

$$\dot{x} = f(t, x, u, v), \qquad x(t_0) = x_0.$$

Denoting $\mathcal{U}$ and $\mathcal{V}$ the sets of measurable functions from $\mathbb{R}$ to U and V respectively, one assumes regularity and growth conditions on f to guarantee existence and uniqueness of the solution $x(\cdot)$ for all (t_0, x_0) and all $(u(\cdot), v(\cdot)) \in \mathcal{U} \times \mathcal{V}$.

2. A termination condition, often given by a target set $\mathcal{T} \in \mathbb{R} \times \mathbb{R}^n$, open or closed according to necessity, defining a final time as $t_1 = \inf\{t \mid (t, x(t)) \in \mathcal{T}\}$. If $\mathcal{T} = \{T\} \times \mathbb{R}^n$, final time is fixed and equal to T. The question of whether there is a finite t_1 is one of central interest in pursuit-evasion games.

3. Sets of admissible closed-loop strategies Φ and Ψ. One should choose them in such a way that replacing (u, v) by a pair $(\phi, \psi) \in \Phi \times \Psi$ in the dynamics always produces a (unique) admissible pair of control functions $(u(\cdot), v(\cdot)) = \Gamma(t_0, x_0; \phi, \psi) \in \mathcal{U} \times \mathcal{V}$.

4. A performance measure, or payoff, typically

$$J(t_0, x_0; u(\cdot), v(\cdot)) = \begin{cases} K(t_1, x(t_1)) + \displaystyle\int_{t_0}^{t_1} L(t, x(t), u(t), v(t))\, \mathrm{d}t & \text{if } t_1 < \infty, \\ \infty & \text{if } t_1 = \infty. \end{cases}$$

We let

$$G(t_0, x_0; \phi, \psi) := J(t_0, x_0; \Gamma(t_0, x_0; \phi, \psi)).$$

5. A concept of "solution," where the first player wants to minimize the performance index while the second one wishes to maximize it. (In our choice of definition of J, we have assumed that player one wants over anything else to make the game terminate. If we define J as the integral even for infinite end-time, Isaacs' tenet of transition may not hold.)

If

$$\inf_{\phi \in \Phi} \sup_{\psi \in \Psi} G(t_0, x_0; \phi, \psi)$$

$$= \sup_{\psi \in \Psi} \inf_{\phi \in \Phi} G(t_0, x_0; \phi, \psi) = V(t_0, x_0),$$

then V is called the Value function of the game. Several concepts of upper Value and lower Value may be defined (including the first and second terms above) that have to coincide for a Value to exist.

Isaacs' Condition In the framework of this short entry, we shall always assume that the game satisfies *Isaacs' condition*. It bears on the *Hamiltonian* $H(t, x, p, u, v) := L(t, x, u, v) + \langle p, f(t, x, u, v) \rangle$ and reads

$$\forall (t, x, p) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n,$$

$$\inf_{u \in \mathsf{U}} \sup_{v \in \mathsf{V}} H(t, x, p, u, v)$$

$$= \sup_{v \in \mathsf{V}} \inf_{u \in \mathsf{U}} H(t, x, p, u, v). \tag{1}$$

Strategies and Value

In pursuit-evasion games, the concept of closed-loop strategies is of the essence, and it is extremely important for all DGs. Yet, allowing state

feedback strategies such as $u(t) = \phi(t, x(t))$, $v(t) = \psi(t, x(t))$, poses a difficult problem: what classes Φ and Ψ of functions ϕ and ψ to allow? The notations $\inf_\phi$ or $\sup_\psi$ have no meaning if one does not answer that question. Experience tells us that discontinuous feedbacks are necessary to find the solution of many examples, but then existence, or uniqueness, of the solution of the dynamical equation cannot be guaranteed.

Isaacs' K-strategies were a partial attempt to address this issue. More developed concepts were proposed, from limit of piecewise constant, or piecewise open-loop, controls (Fleming 1961; Friedman 1971) to extensions of the notion of solution of a differential equation (Krasovskii and Subbotin 1977), also proving the existence of a Value. The equivalence of all these Values was an issue until the advent of viscosity solutions of Isaacs' equation.

A tool used to accommodate state-feedback strategies (Bernhard 1977) is

Lemma 1 (Berkovitz) *If $\mathcal{V} \subset \Psi$, then, $\forall \phi$ for which this expression is well defined,*

$$\sup_{\psi \in \Psi} G(t_0, x_0; \phi, \psi) = \sup_{v(\cdot) \in \mathcal{V}} G(x_0, t_0, \phi, v(\cdot)).$$

As a consequence, a saddle-point $(\phi^\star, \psi^\star)$ solution is defined by

$$\forall u(\cdot) \in \mathcal{U}, \forall v(\cdot) \in \mathcal{V}$$
$$G(t_0, x_0; \phi^\star, v(\cdot)) \leq V(t_0, x_0)$$
$$\leq G(t_0, x_0; u(\cdot), \psi^\star), \qquad (2)$$

confronting the closed-loop saddle point strategies to open-loop controls only. (This proves useful in the analysis of Nash equilibria of nonzero-sum DGs.)

Another consequence of Berkovitz' lemma is that if a DG has a saddle point in open-loop controls, it is a saddle point over closed-loop controls as well. (But the existence condition may be less stringent for the later.) The relationship between different forms of the strategies has been

further clarified by Başar (1977) and Başar and Olsder (1982).

As far as the existence of the Value is concerned, the problem for a large class of DGs is solved with *non-anticipative strategies* defined as $\Phi : \mathcal{V} \to \mathcal{U}$ such that

$$\forall t, \ [\forall s < t \quad v_1(s) = v_2(s)] \Rightarrow [\phi(v_1(\cdot))(t)$$
$$= \phi(v_2(\cdot))(t)],$$

and likewise for Ψ (notice that for this concept of strategies, (2) is the natural formulation of a saddle point) and with the notion of *viscosity solution of Isaacs' equation*. See Theorem 1 below.

Games of Pursuit Evasion

An important class of DGs is the game of pursuit evasion. Typically, in these games the state x is composed of a sub-vector y of *Pursuer state(s)* and a sub-vector z of *Evader state(s)*. The dynamical function f is separated likewise, the dynamics of the Pursuer depending on the *Pursuer's control(s)* and that of the Evader on the *Evader's control(s)*. Typically, the payoff is time until *capture* defined as $(t, x(t)) \in \mathcal{T}$ (the target is often called *capture set*). This form of DG automatically satisfies Isaacs' condition (1).

Qualitative Game
In pursuit-evasion games, the main issue is to distinguish initial states, called *capturable*, for which a Pursuer's strategy causing finite-time capture against any defense exists, from those, called *safe*, for which the Evader has a strategy guaranteeing *escape* against any defense. This is the topic of the *qualitative game* or *game of kind* (Isaacs). A *theorem of the alternative* is one which states that for a particular (class of) game(s), every initial state is either capturable or safe. Such theorems have been proved for classes of pursuit-evasion games covering essentially all cases of interest, under Isaacs condition (1) with $L = 0$ (Cardaliaguet 1996; Cardaliaguet et al. 2001; Krasovskii and Subbotin 1977).

Capturable states are separated from safe states by a *barrier*, a piecewise smooth manifold which has to be *semipermeable*. This means that for all (t, x) on the barrier where this barrier is a smooth manifold with normal $v(t, x)$, it should hold that

$$\min_{u \in U} \max_{v \in V} \langle v(t, x), f(t, x, u, v) \rangle$$
$$= \max_{v \in V} \min_{u \in U} \langle v(t, x), f(t, x, u, v) \rangle = 0 \, .$$

A minimax pair $(u, v) = (\hat{\varphi}(t, x, v), \hat{\psi}(t, x, v))$ is called a pair of semipermeable strategies. If the boundary of the capture set is a smooth manifold with local outward normal $n(t, x)$, its *usable part* is the region where $\inf_{u \in U} \sup_{v \in V} \langle n(t, x), f(t, x, u, v) \rangle < 0$. The *natural barrier* is a semipermeable manifold constructed backward from its boundary (the BUP), with n as final v and using the characteristic equations:

$$\dot{x} = f(x, \hat{\phi}, \hat{\psi}) \, , \qquad \dot{v}^t = -v^t \frac{\partial f(t, x, \hat{\varphi}, \hat{\psi})}{\partial x} \, .$$

(These trajectories are *abnormal trajectories* of the calculus of variations). In most examples, only part of the manifold thus constructed is a barrier, and the complete barrier is made of manifolds pieced together according to a junction condition insuring that the corners "do not leak" (Breakwell), analogous to the corner conditions of the next section.

Quantitative Game

The quantitative game, or *game of degree* (Isaacs), is played inside the capture zone, typically with time of capture as the payoff. It is ruled by Isaacs' equation in a fashion similar to that of games of finite duration (see below). Yet, the interplay between the qualitative and the quantitative game may be quite subtle and plays a prominent role in determining the actual capture zone. The Value function is usually discontinuous across other barriers inside the capture zone.

Other Approaches

Other approaches have been developed to solve games of pursuit evasion.

An early approach by Pontryagin (1967), extended by Pshenichnyi (1968), used geometric methods for linear pursuit-evasion games with convex compact control sets. Krasovskii's *stable bridges* (Krasovskii and Subbotin 1977) are a concept close to Isaacs' semipermeability. Patsko and Turova (2001) have developed, for some families of DGs, an efficient numerical procedure to compute recursively hypersurfaces of constant time-to-capture, whose discontinuities display the barriers. Cardaliaguet et al. (1999) have developed a theoretical and numerical procedure building on Aubin's viability theory, which requires less regularity on the data than other approaches.

Provided that care be applied, a quantitative game may be transformed into a family of qualitative games – an approach used by Krasovskii, Blaquière et al., and Cardaliaguet et al. – and conversely, a fruitful approach is to investigate capturability of initial states as a function of a parameter defining the "size" of the capture set, imbedding the qualitative game into a quantitative game of the type *game of approach*.

Games of Finite Duration

Wherever termination of the game is not an issue, the major tool in investigating a DG is Isaacs' equation, a partial differential equation bearing on the Value function:

$$\forall (t, x) \notin \mathcal{T} \, , \quad \frac{\partial V}{\partial t}(t, x)$$
$$+ \min_{u \in U} \max_{v \in V} H(t, x, \nabla_x V, u, v) = 0 \, ,$$
$$\forall (t, x) \in \mathcal{T} \, , \quad V(t, x) = K(t, x) \, .$$
$$(3)$$

For any DG where all trajectories are transverse to the boundary $\partial \mathcal{T}$, and with adequate regularity conditions on the data (and still under condition (1)), it holds that

Theorem 1 *The DG has a Value in non-anticipative strategies, which is the only bounded,*

uniformly continuous viscosity solution of the equation obtained by changing signs in (3) as $-\partial V/\partial t - \min_{u \in \cup} \max_{v \in V} H = 0$. *And all other Values coincide.*

One possible way to solve Isaacs' equation is via the investigation of its field of characteristics. Their equations are Isaacs' *retrograde path equations*: let $(\hat{u}, \hat{v}) = (\hat{\varphi}(t, x, p), \hat{\psi}(t, x, p))$ be the saddle point of $H(t, x, p, u, v)$, assumed here to be unique, one integrates from the target set backward:

$$\dot{x} = f(t, x, \hat{u}, \hat{v}), \tag{4}$$

$$\dot{p} = -\left(\frac{\partial H(t, x, p, \hat{u}, \hat{v})}{\partial x} \right)^t. \tag{5}$$

The above equations are similar to Pontryagin's maximum principle equations. However, a major difference lies in the *corner conditions*. While Pontryagin's theorem extends to control theory the Erdman-Weierstrass condition stating that the adjoint vector (here p) is continuous along an extremal trajectory, in (4) and (5), p is to coincide with $\nabla_x V$ and **may be discontinuous along an extremal trajectory**. These discontinuities cannot be found by a local analysis along an isolated trajectory and require that a complete field of extremals be constructed, synthesizing a state feedback strategy.

The analysis of the conditions that hold at such corners, equivocal manifolds (Isaacs), envelope manifolds (Breakwell), and focal manifolds (Merz), has been a large part of the early Isaacs-Breakwell theory. It has been for its larger part synthesized by Bernhard (1977), except a general constructive analysis of focal manifolds which had to wait until Melikyan and Bernhard (2005).

The absence of a "two-sided Pontryagin principle" for closed-loop differential games forces one to resort to the solution of Isaacs' equation or an equivalent. This is the reason why no practical method of solution exists beyond a state dimension of 3 or 4, counting time if the game is not time invariant. An exception is the linear quadratic game. (See article ▶ Linear Quadratic Zero-sum Two-person Differential Games in this encyclopaedia).

Conclusion

Except for very particular games, "solving" a DG remains a difficult task. Numerical methods suffer the famous "curse of dimensionality." Moreover, many of them strive to compute the Value function. But the optimal strategies typically depend on the gradient of the Value function, requiring a stronger convergence of the approximation algorithms than pointwise, or C^0 or L^2, if they are to be computed as well. Further advances in numerical algorithms tackling this problem would be useful, as well as uncovering new classes of DGs for which further analytical results could be obtained.

Cross-References

▶ Dynamic Noncooperative Games
▶ Game Theory: Historical Overview
▶ Linear Quadratic Zero-Sum Two-Person Differential Games

Bibliography

Başar T (1977) Informationally nonunique equilibrium solutions in differential games. SIAM J Control Optim 15:636–660

Başar T, Olsder GJ (1982) Dynamic noncooperative game theory. Academic, London/New York

Bernhard P (1977) Singular surfaces in differential games, an introduction. In: Haggedorn P, Olsder GJ, Knobloch H (eds) Differential games and applications. Lecture notes in information and control sciences, vol 3. Springer, Berlin, pp 1–33

Blaquière A, Gérard F, Leitmann G (1969) Quantitative and qualitative games. Academic, New York

Breakwell JV, Merz AV (1969) Toward a complete solution of the homicidal chauffeur game. In: Ho Y-C, Leitmann G (eds) Proceedings of the first international conference on the theory and applications of differential games, Amherst

Breakwell JV (1977) Lecture notes. In: Hagedorn P, Knobloch HW, Olsder G-J (eds) Theory and applications of differential games. Springer lecture notes in control and information sciences, vol 3. Springer, Berlin, pp 70–95

Cardaliaguet P (1996) A differential game with two players and one target. SIAM J Control Optim 34:1441–1460

Cardaliaguet P, Quincampoix M, Saint-Pierre P (1999) Set-valued numerical methods for optimal control and differential games. In: Nowak A (ed) Stochastic and differential games. Theory and numerical methods. Annals of the international society of dynamic games. Birkhaüser, Boston, pp 177–247

Cardaliaguet P, Quincampoix M, Saint-Pierre P (2001) Pursuit differential games with state constraints. SIAM J Control Optim 39:1615–1632

Crandal MG, Lions P-L (1983) Viscosity solutions of Hamilton Jacobi equations. Trans Am Math Soc 277:1–42

Elliot RJ, Kalton NJ (1972) The existence of value in differential games of pursuit and evasion. J Differ Equ 12:504–523

Evans LC, Souganidis PE (1984) Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. Indiana Univ Math J 33:773–797

Fleming WK (1961) The convergence problem for differential games. Math Anal Appl 3:102–116

Friedman A (1971) Differential games. Wiley, New York

Isaacs R (1965) Differential games, a mathematical theory with applications to optimization, control and warfare. Wiley, New York

Isaacs RP (1951) Games of pursuit. Technical report P-257, The Rand corporation

Krasovskii N, Subbotin A (1977) Jeux différentiels. MIR, Moscow

Lions P-L (1982) Generalized solutions of Hamilton-Jacobi equations. Pitman, Boston

Lions P-L, Souganidis PE (1985) Differential games, optimal control, and directional derivatives of viscosity solutions of Bellman's and Isaacs' equations. SIAM J Control Optim 23:566–583

Melikyan A, Bernhard P (2005) Geometry of optimal trajectories around a focal singular surface in differential games. Appl Math Optim 52:23–37

Patsko VS, Turova VL (2001) Level sets of the value function in differential games with the homicidal chauffeur dynamics. Int Game Theory Rev 3:67–112

Pontryagin LS (1967) Linear differential games I and II. Soviet Math Doklady 8:769–771, 910–912

Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishenko EF (1962) The mathematical theory of optimal processes. Wiley, New York

Pshenichnyi BN (1968) Linear differential games. Autom Remote Control 29:55–67

Quincampoix M (2009) Differential games. In: Meyers N (ed) Encyclopaedia of complexity and system science. Springer, New York, pp 1948–1956

Roxin EO (1969) The axiomatic approach in differential games. J Optim Theory Appl 3(3):153–163

Varaiya P, Lin Y (1969) Existence of saddlepoint in differential games. SIAM J Control 7: 141–157

Q

QFT

▶ Quantitative Feedback Theory

Quantitative Feedback Theory

Mario Garcia-Sanz
Case Western Reserve University, Cleveland, OH, USA

Synonyms

QFT

Abstract

Designing reliable and high-performance control systems is an essential priority of every control engineering project. In many practical circumstances the presence of model uncertainty challenges the design. One robust control approach for these cases, deeply rooted in the classical frequency domain, is quantitative feedback theory (QFT). Providing a control solution that guarantees the achievement of a multi-objective set of performance specifications for every plant within the model uncertainty (quantification), QFT balances the trade-off between the simplicity of the compensator

structure and the minimization of the activity of the controller at each frequency ("cost of feedback"). Previous results indicate that the QFT methodology has been able to provide successful control solutions to a large variety of real applications, including linear and non-linear plants, stable and unstable systems, multi-input multi-output processes, minimum and non-minimum phase plants, containing time-delay, lumped or distributed parameters, etc.

Keywords

Frequency domain control; Quantitative controller design; Robust control

Definition

Quantitative Feedback Theory (QFT) is a robust control engineering design methodology that uses the feedback to simultaneously and quantitatively: (1) reduce the effects of *plant uncertainty* and (2) satisfy *performance control specifications*. The method searches for a controller that guarantees the satisfaction of the required performance specifications for every plant within the model uncertainty (*robust control*).

QFT is rooted in the classical frequency domain. It involves Bode diagrams and Nichols charts (magnitude/phase diagrams). It relies on the observation that feedback is needed when

the plant presents model uncertainty and/or there are uncertain disturbances. QFT balances quantitatively: (a) the simplicity of the controller structure, (b) the minimization of the so-called *cost of feedback*, controller magnitude at each frequency, (c) the plant model uncertainty and (d) the achievement of the desired performance specifications, all at each frequency of interest. The technique has been successfully applied to a wide variety of real-world control problems.

Historical Notes

Many of the frequency domain fundamentals were established by Hendrik Bode in his seminal book *Network Analysis and Feedback Amplifier Design*, published in 1945 (Van Nostrand). The book strongly influenced the understanding of automatic control theory for many years, especially where system sensitivity and feedback constraints are concerned.

Almost 20 years later, in 1963, a new influential book entitled *Synthesis of Feedback Systems* (Academic Press), written by Isaac Horowitz, proposed for the first time a formal combination of the frequency domain methodology with plant model uncertainty (*robust control*) under a quantitative analysis. The new book addressed an extensive set of sensitivity problems in feedback control and was the first work in which a control problem was treated quantitatively in a systematic way. The book laid the foundation

for a new control design methodology that had been introduced briefly in a previous paper by Horowitz in 1959: the one that became known as *Quantitative feedback theory* (or QFT) in the early 1970s.

Fundamentals

A detailed study of the *QFT fundamentals and applications* can be found in the books written by Garcia-Sanz and Houpis (2012), Houpis et al. (2006), Sidi (2002), Yaniv (1999), and Horowitz (1993); see the "Recommended Reading" section.

The QFT methodology provides a multi-criteria engineering understanding of the controller design process, as it quantifies the balance among the controller structure, cost of feedback, performance specifications, and model plant uncertainty at each frequency of interest. The basic steps of the QFT methodology are summarized in Fig. 1 and are presented in the following sub-sections.

Define Plant Model and Uncertainty: Templates Generation (Steps 1, 2 & 3)

First of all, the dynamics of the plant to be controlled are described in the frequency domain. Taking the plant model in terms of transfer functions with mixed parametric, non-parametric and even model structure uncertainty, the frequency domain description is carried out
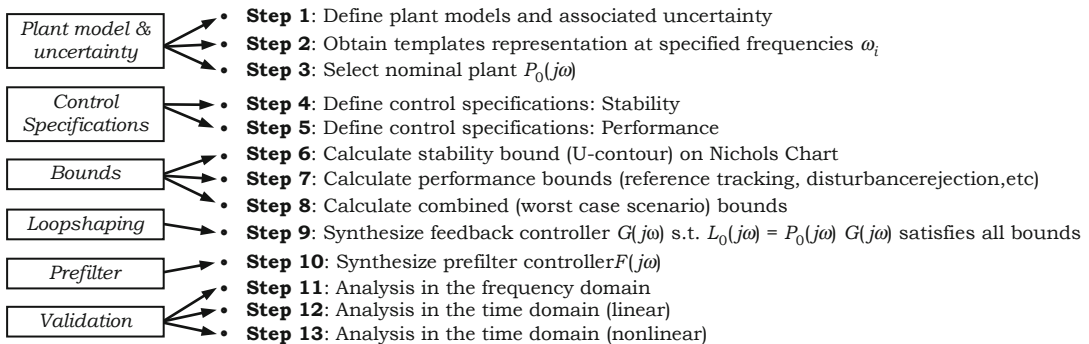
Quantitative Feedback Theory, Fig. 1 Summary of QFT controller design methodology
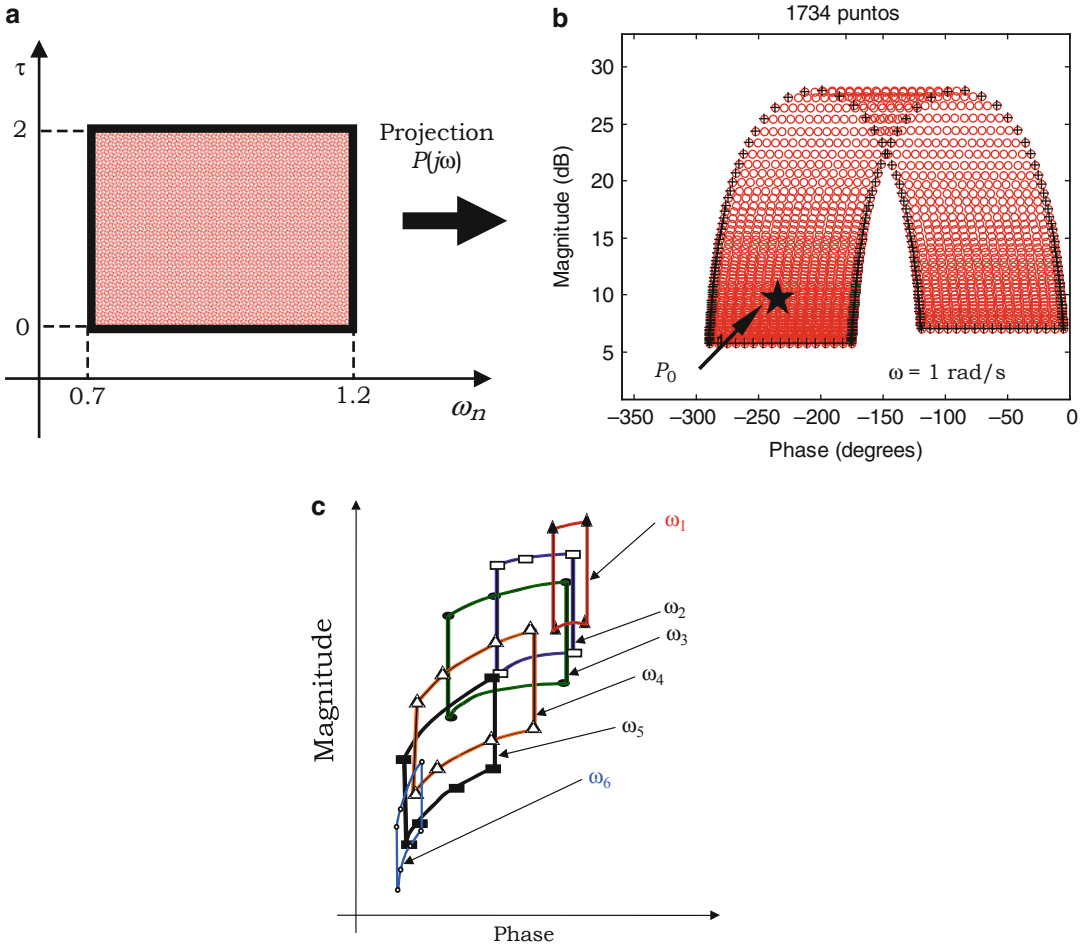
Quantitative Feedback Theory, Fig. 2 From the *parameter space* to the *Nichols chart*: (**a**) 2-dimensional parameter space, (**b**) Template of the plant $P(j\omega)$ at $\omega = 1$ rad/s, (**c**) Typical templates for frequencies $\omega \in [\omega_{\min}, \omega_{\max}]$

by calculating "*templates*", which are sets of complex numbers at each frequency of interest $\omega \in [\omega_{\min}, \omega_{\max}]$ rad/second: a projection of the n-dimensional parameter space through the transfer function/functions onto the Nichols chart.

As an example, and for "$\omega = 1$ rad/s, Fig. 2b represents the QFT template of the 3-parameter plant $P(j\omega) = exp(-j\omega\,\tau)\Big/ \Big((j\omega)^2 + 2\,\zeta\,\omega_n\,(j\omega) + \omega_n^2\Big)$, with $\omega_n \in [0.7, 1.2]$, $\tau \in [0, 2]$, and $\zeta = 0.02$.

Each template $\Im P(j\omega_i) = \{P(j\omega_i)\}$ represents on the Nichols chart and at a specific frequency ω_i all the possible plants within the model uncertainty (see Fig. 2c). One particular case, defined as a set of specific parameters of the n-dimensional parameter space is arbitrarily selected to define the nominal plant $P_0(j\omega)$, a member of the family of plants within the uncertainty (see Fig. 2b).

Define Control Specifications (Steps 4 & 5)

The standard two-degree-of-freedom (2DOF) control system diagram is shown in Fig. 3.

Quantitative Feedback Theory, Fig. 3 Multi-Input-Single-Output 2DOF feedback control system. $s \sim j\omega$
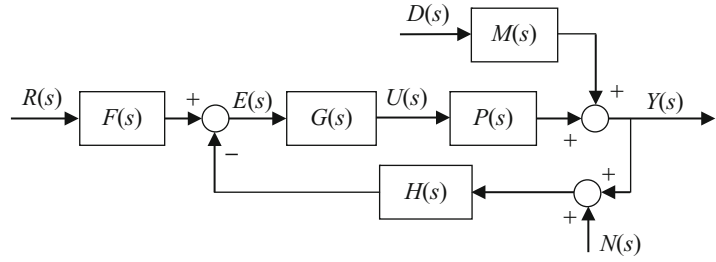
It includes the set of uncertain plants $P(j\omega)$ to be controlled, the disturbance dynamics $M(j\omega)$, the feedback path dynamics $H(j\omega)$, and the loop controller $G(j\omega)$ and prefilter $F(j\omega)$, both to be design. On the other hand, $R(j\omega)$, $E(j\omega)$, $U(j\omega)$, $Y(j\omega)$, $D(j\omega)$, and $N(j\omega)$ are vectors representing respectively the reference input, signal error, controller output, plant output, disturbance input, and sensor noise input. From the diagram (Fig. 3), it is easy to derive the following three input/output equations (note the dependency on $j\omega$ is removed):

$$Y = \frac{PG}{1+PGH}FR + \frac{M}{1+PGH}D - \frac{PGH}{1+PGH}N;$$

$$U = \frac{G}{1+PGH}FR - \frac{GH}{1+PGH}(MD+N) \text{ and}$$

$$E = \frac{1}{1+PGH}FR - \frac{HM}{1+PGH}D - \frac{H}{1+PGH}N$$

Without losing generality, and with a straightforward block diagram manipulation, $F(s)$ and $G(s)$ can be modified to have $H(s) = 1$. Now, the stability and performance specifications are defined by limiting the magnitude of each transfer function of the three previous equations at each frequency of interest, $|T_k(j\omega)| \leq \delta_k(\omega)$, $k = 1$–4, such that,

Stability and noise reduction: $|T_1(j\omega)| =$
$$\left|\frac{Y(j\omega)}{R(j\omega)\,F(j\omega)}\right| = \left|\frac{Y(j\omega)}{N(j\omega)}\right| = \left|\frac{P(j\omega)\,G(j\omega)}{1+P(j\omega)\,G(j\omega)}\right|$$
$\leq \delta_1(\omega), \quad \omega \in \Omega_1,$

Disturbance rejection: $|T_2(j\omega)| = \left|\frac{Y(j\omega)}{D(j\omega)}\right| =$
$$\left|\frac{M(j\omega)}{1+P(j\omega)\,G(j\omega)}\right| \leq \delta_2(\omega), \quad \omega \in \Omega_2,$$

Control effort reduction: $|T_3(j\omega)| =$
$$\left|\frac{U(j\omega)}{M(j\omega)\,D(j\omega)}\right| = \left|\frac{U(j\omega)}{N(j\omega)}\right| = \left|\frac{U(j\omega)}{R(j\omega)\,F(j\omega)}\right| =$$
$$\left|\frac{G(j\omega)}{1+P(j\omega)\,G(j\omega)}\right| \leq \delta_3(\omega), \quad \omega \in \Omega_3$$
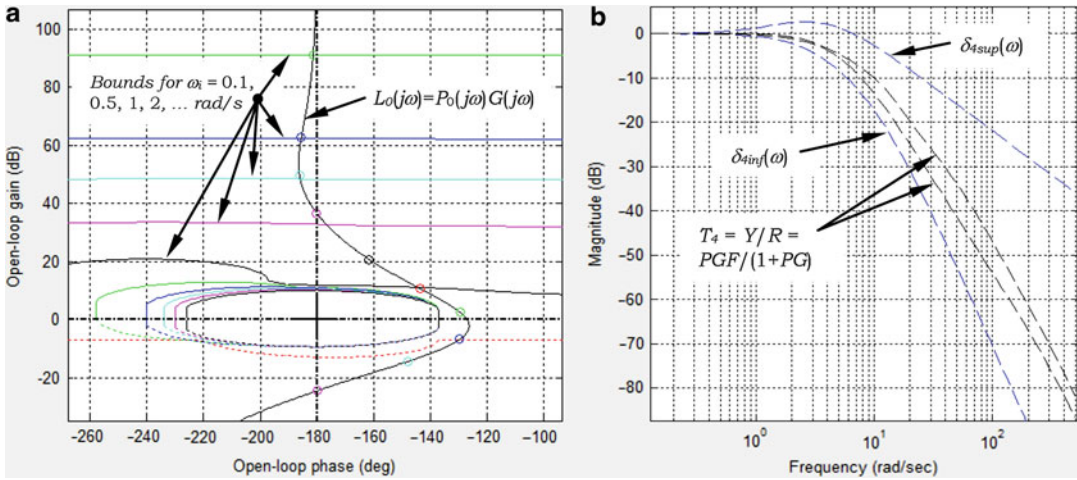
Reference tracking: $\delta_{4\text{inf}}(\omega) < |T_4(j\omega)| =$
$$\left|\frac{Y(j\omega)}{R(j\omega)}\right| = \left|F(j\omega)\,\frac{P(j\omega)\,G(j\omega)}{1+P(j\omega)\,G(j\omega)}\right| \leq$$
$\delta_{4\text{sup}}(\omega), \quad \omega \in \Omega_4,$

$$\frac{|G(j\omega)\,P_d(j\omega)|}{|G(j\omega)\,P_e(j\omega)|}\,\frac{|1+G(j\omega)\,P_e(j\omega)|}{|1+G(j\omega)\,P_d(j\omega)|} \leq \delta_4(\omega)$$
$$= \frac{\delta_{4\sup}(\omega)}{\delta_{4\inf}(\omega)}, \quad \omega \in \Omega_4$$

QFT Bounds (Steps 6, 7 & 8)

For the nominal plant $P_0(j\omega)$, the QFT methodology converts the stability and performance specifications $\delta_k(\omega)$ and the model plant uncertainty into a set of constrains or bounds for each frequency of interest on the Nichols chart (the *Horowitz-Sidi Bounds*).

The ω_i plant template, $\Im P(j\omega_i) = \{P(j\omega_i)\}$, is approximated by a finite set of plants $\{P_r(j\omega_i), r = 1, 2 \ldots\}$. Each plant can be expressed in its polar form as $P_r(j\omega_i) = p(\omega_i)\,exp(j\theta(\omega_i)) = p\angle\theta$. Likewise the controller polar form is $G(j\omega_i) = g(\omega_i)\,exp(j\phi) = g\angle\phi$, with a controller phase ϕ that varies from -2π to 0. Therefore, and for every frequency ω_i, the previous control specifications $\{|T_k(j\omega_i)| \leq \delta_k(\omega_i), k = 1, \ldots, 4\}$ are translated into a set of quadratic inequalities with the format $I_{\omega_i}^k(p, \theta, \delta_k, \phi) = a\,g^2 + b\,g + c \geq 0$, such that,

Quantitative Feedback Theory, Fig. 4 (**a**) QFT-bounds and $G(j\omega)$ design –loopshaping–. (**b**) Prefilter $F(j\omega)$ design: reference tracking specifications $\delta_{4\text{sup}}(\omega)$ and $\delta_{4\text{inf}}(\omega)$, and *upper* and *lower* limits of T_4 due to the plant uncertainty: $\delta_{4\text{inf}}(\omega) \leq |T_4| \leq \delta_{4\text{sup}}(\omega)$

Stability and noise reduction: $p^2 \left(1 - \frac{1}{\delta_1^2}\right)$
$g^2 + 2\,p\,\cos(\phi + \theta)\,g + 1 \geq 0,$

Disturbance rejection: $p^2\,g^2 + 2\,p\,\cos(\phi + \theta)$
$g + \left(1 - \frac{m^2}{\delta_2^2}\right) \geq 0,$ with typically $m = p, 1,$
or other options,

Control effort reduction: $\left(p^2 - \frac{1}{\delta_3^2}\right)\,g^2 +$
$2\,p\,\cos(\phi + \theta)\,g + 1 \geq 0,$ and

Reference tracking: $p_e^2 p_d^2 \left(1 - \frac{1}{\delta_4^2}\right)\,g^2 +$
$2\,p_e\,p_d \left(p_e\,\cos(\phi + \theta_d) - \frac{p_d}{\delta_4^2}\,\cos(\phi + \theta_e)\right) g$
$+ \left(p_e^2 - \frac{p_d^2}{\delta_4^2}\right) \geq 0$

Now, with an appropriate algorithm (see references), the above quadratic inequalities are translated into a set of curves on the Nichols chart for each frequency of interest and type of specification: the *individual specification bounds*. Then, the more demanding (worst case) bound, i.e., the most restrictive one at every phase and each frequency of interest is computed to obtain the intersection of bounds, or the *combined QFT bounds* (see Fig. 4a).

Controller $G(j\omega)$ Design: Loop-Shaping (Step 9)

Although the objective of designing a controller for an infinite number of plants seems to be a very arduous task (there is an infinite number of plants due to the model uncertainty), the integration of all the information (uncertainty and specifications) in a set of simple curves (the QFT bounds) will allow the designer to use just a single plant, the nominal plant P_0, and the bounds to design the controller.

Then, in the design stage (loop-shaping), the controller $G(j\omega)$ is synthesized on the Nichols chart by adding poles and zeros until the nominal loop, defined as $L_0(j\omega) = P_0(j\omega)G(j\omega)$, lies near its bounds (see Fig. 4a). The bounds express the plant models with uncertainty and the performance specifications at each frequency. An optimal controller in the sense of QFT will be obtained if $L_0(j\omega)$ lies exactly on the bounds at each frequency. Practically speaking, a good design will place $L_0(j\omega)$ above the continuous-line bounds and below the dashed-line bounds, and will have the minimum possible magnitude at every frequency. A general formulation for the controller structure $G(s)$ is expressed by the following transfer function:

$$G(s) = \frac{k_G \prod\limits_{i=1}^{n_{rz}} \left(\frac{s}{z_i} + 1\right) \prod\limits_{i=1}^{n_{cz}/2} \left(\frac{s^2}{|z_i|^2} + \frac{2\,Re(z_i)}{|z_i|^2}s + 1\right)}{s^r \prod\limits_{j=1}^{m_{rp}} \left(\frac{s}{p_j} + 1\right) \prod\limits_{j=1}^{m_{cp}/2} \left(\frac{s^2}{|p_j|^2} + \frac{2\,Re(p_j)}{|p_j|^2}s + 1\right)}$$

where k_G is the controller gain, z_i is a zero (real or complex) with m_{rz} and m_{cz} the number of real and complex zeroes respectively, and p_j is a pole (real or complex) with m_{rp} and m_{cp} the number of real and complex poles respectively (m_{cz} and m_{cp} even numbers). The controller may have also some poles at the origin (integrators), with $r = 0$, 1 or 2, etc.

Prefilter $F(j\omega)$ Design (Step 10)

If the feedback system includes a reference tracking problem, then the best choice is to use a prefilter $F(s)$ – the second degree of freedom. While the feedback controller $G(s)$ reduces the effect of the uncertainty and improves stability, disturbance rejection, and other specifications, the prefilter $F(s)$ is designed to fulfill reference tracking requirements. Figure 4b shows a typical prefilter design in the Bode diagram. $\delta_{4\sup}(\omega)$ and $\delta_{4\inf}(\omega)$ are the reference tracking specifications, defined as a band (outer dashed lines, Fig. 4b). The transfer function T_4 shows an upper and a lower limit (inner dashed lines, Fig. 4b) due to the plant uncertainty. After an appropriate prefilter design, the T_4 limits will be in the middle of the $\delta_{4\sup}$–$\delta_{4\inf}$ band:

$$\delta_{4\inf}(\omega) \le |T_4| \le \delta_{4\sup}(\omega), |T_4(j\omega)| = \left|\frac{Y(j\omega)}{R(j\omega)}\right|$$

$$= \left|\frac{P(j\omega)\,G(j\omega)}{1 + P(j\omega)\,G(j\omega)}\,F(j\omega)\right|$$

Validation (Steps 11, 12, 13)

Once the design of the controller (and prefilter if needed) is finished, it will be convenient to analyze the performance of the complete control system under different scenarios, including: (a) frequency domain analysis of each specification for all the significant plants within the model uncertainty and (b) time domain simulations, typically using a Monte Carlo campaign for the uncertainty, first with the linear system and then with nonlinear elements (saturation, etc.).

Programs and Data

Computer-aid-design (CAD) tools have definitely facilitated the use of QFT. The MATLAB code of the interactive object-oriented QFT CAD tool developed by Garcia-Sanz et al. for ESA-ESTEC (2014) can be found at http://cesc.case.edu/OurQFTCT.htm (free download). Another popular QFT CAD tool in the 1990s, developed by Borghesani, Chait & Yaniv, can be found at http://www.terasoft.com/products/QFT/index.html.

Applications and Future Directions

- QFT has been successfully applied to a wide variety of control problems, including stable and unstable plants minimum and non-minimum phase systems, single-input single-output and multiple-input multiple-output processes, with linear and nonlinear characteristics, longtime delay, distributed parameter systems, and time-varying plants; and has been combined with feed-forward control topologies, multi-loop systems, etc. Also, QFT has been used in many real-world applications: e.g., flight control, wind energy, water treatment plants, spacecraft, power systems, mechanical systems, motion control, chemical reactors, etc. (see Garcia-Sanz and Houpis 2012; Houpis et al. 2006).
- Future research on QFT includes among others new multiple-input multiple-output techniques, nonlinear plants, distributed parameter systems, load-sharing control, etc.

Cross-References

▶ Classical Frequency-Domain Design Methods
▶ Polynomial/Algebraic Design Methods
▶ Robust Adaptive Control
▶ Spectral Factorization

Bibliography

Garcia-Sanz M, Houpis CH (2012) Wind energy systems: control engineering design. Part I: QFT control. Part II: wind turbines control with QFT. CRC/Taylor & Francis, Boca Raton, Florida

Garcia-Sanz M, Mauch A, Philippe C (2014) The QFT control toolbox (QFTCT): an interactive object-oriented Matlab CAD tool for QFT robust control systems design. European space agency ESA-ESTEC, Public University of Navarra, Case Western Reserve University, 2008–2014. Free download at http://cesc.case.edu/OurQFTCT.htm

Horowitz IM (1959) Fundamental theory of automatic linear feedback control systems. IRE Trans Autom Control 4:5–19

Horowitz I (1993) Quantitative feedback design theory (QFT). QFT Publications, Denver, Colorado

Houpis CH, Rasmussen SJ, Garcia-Sanz M (2006) Quantitative feedback theory: fundamentals and applications, 2nd edn. CRC/Taylor and Francis, Boca Raton, Florida

Sidi M (2002) Design of robust control systems: from classical to modern practical approaches. Krieger Publishing, Malabar, Florida

Yaniv O (1999) Quantitative feedback design of linear and non-linear control systems. Kluver Academic, Boston, Massachusetts

Quantized Control and Data Rate Constraints

Girish N. Nair
Department of Electrical & Electronic Engineering, University of Melbourne, Melbourne, VIC, Australia

Abstract

This article briefly describes the topic of quantized control with limited data rates. The focus is on the problem of stabilizing a linear time-invariant plant over a digital channel and the associated *data rate theorems*. It is shown that the deepest results in this area require a unified treatment of its communications and control aspects.

Keywords

Control under communication constraints; Quantization; Quantized control

Introduction

One of the standard assumptions of classical control theory is that the signals sent from sensors to controllers and from controllers to actuators take continuous values with infinite precision. The advent of computer-based and digitally networked control systems challenged this assumption, since the analog plant outputs or control variables in such systems must be reduced to finite bit strings or discrete symbols for storage, manipulation, and transmission. This process of converting a continuous-valued variable into a finite-valued one is called *quantization* and entails a potentially significant loss of resolution and closed-loop performance. Quantized control is concerned with the analysis and design of control systems which feature such analog-to-digital conversions in the feedback loop.

There is a vast literature on this topic and the aim of this article is to briefly explain some of its key ideas. For reasons of space, the discussion is largely confined to the question of how to stabilize a linear time-invariant plant over a digital channel. It is shown that the deepest results here emerge from treating the communications and control aspects jointly, instead of separately. The reader is referred to the survey (Nair et al. 2007) and the references therein for a discussion of other issues such as optimality and transient performance.

Q

Quantization

Quantization has long been an object of study in communications and information theory – see Gersho and Gray (1993) and the references therein. In its simplest form, a signal $x(\cdot) : \mathbb{R} \to \mathbb{R}^n$ is first *sampled* at regular time intervals $t = 0, \tau, 2\tau, \ldots$ to yield a discrete-time signal $(x(k\tau))_{k \in \mathbb{Z}}$, with the sampling frequency $1/\tau$ chosen to be greater than the Nyquist frequency of x (i.e., twice its bandwidth). Each sample $x_k := x(k\tau)$ is then passed through a static, memoryless quantizer Q to yield a quantized discrete-time signal

$$x_k^q = Q(x_k) \in \{q^1, \ldots q^M\} \subset \mathbb{R}^n, \quad k \in \mathbb{Z}_{\geq 0}. \tag{1}$$

which can take M distinct values in $\mathbb{R}^n$. If the quantizer is known to both transmitter and receiver, each of these M values can be represented by a binary string with $\lceil \log_2 M \rceil$ bits. When the input dimension $n = 1$, the quantizer is called scalar; otherwise, it is a vector quantizer. The regions $R^i := Q^{-1}(q^i), 1 \leq i \leq M$, are called the quantizer cells and together form a partition of $\mathbb{R}^n$. Thus an M-valued quantizer is fully defined by its quantizer cells R^i and associated quantizer points $q^i, 1 \leq i \leq M$.

The quantization error or *quantizer noise* is defined as $n_k := x_k^q - x_k$. When the inputs x_k are identically distributed random variables, then a standard goal is to design Q so as to minimize the mean-square quantizer noise

$$D := \mathrm{E}[||Q(x_k) - x_k||^2], \tag{2}$$

where $\mathrm{E}[\cdot]$ is the expectation functional. This yields an optimal quantizer Q_* with cells that satisfy the nearest-neighbor property, i.e.,

$$x \in R_*^i \Rightarrow ||Q_*(x) - q_*^i|| \leq ||Q_*(x) - q_*^j||, \forall j \neq i.$$

When $|| \cdot ||$ is the Euclidean norm (possibly weighted), the quantizer cells $R_*^i, 1 \leq i \leq m$, are convex polygons and form a *Voronoi partition* of $\mathbb{R}^n$, and furthermore q_*^i is the centroid of R_*^i with respect to the stationary distribution F_X of

x_k, i.e., $q_*^i = \mathrm{E}[x_k | x_k \in R_*^i]$. As a consequence, the optimal quantizer is statistically unbiased, i.e., $\mathrm{E}[n_k] = 0$, and furthermore x_k and the quantizer noise n_k are uncorrelated at time k, i.e., $\mathrm{E}[x_k n_k^\mathrm{T}] = 0$. However, note that n_k and x_j may be correlated for $j \neq k$, and (n_k) may itself be a correlated process.

If Q is not optimal but M is large (i.e., the quantizer is *high resolution* or *fine*), then $\mathrm{E}[x_k n_k^\mathrm{T}] = \mathrm{o}(1/M)$, provided that q^i is the centroid of R^i with respect to Lebesgue measure μ and x_k has a probability density function (pdf) f_X with suitable continuity properties. The reasoning here is that each region R_i will typically be very small, so that f_X will not vary much on each R^i, yielding a conditional pdf of x_k given R_i that is approximately uniform on R_i.

When Q is a scalar uniform quantizer on an interval $[a, b]$, these considerations yield the asymptotic formula

$$D \approx (b - a)^2/(12M^2), \tag{3}$$

provided that the overload regions – i.e., the tails of $f_X(x)$ on the regions $x < a$ or $x > b$ – make negligible contributions to D. Note that this expression does not depend on the distribution of the input. For large M, it can be shown that the optimal vector quantizer has a normalized point density proportional to $f_X^{1/3}$ and yields

$$D_{\min} \approx \frac{c}{M^2} \left(\int f_X(x)^{1/3} \mathrm{d}\mu(x) \right)^3, \tag{4}$$

where the constant c depends only on n.

Quantized Control: Basic Formulation

Much of the theory of quantized control concerns finite-dimensional linear time-invariant (LTI) plants. A formulation is provided in this section to help fix ideas, for the case of a single feedback loop containing a single errorless digital channel.

Consider the discrete-time plant

$$x_{k+1} = Ax_k + Bu_k + v_k, \quad y_k = Fx_k + w_k, \tag{5}$$

where at every time $k \in \mathbb{Z}_{\geq 0}$, $x_k \in \mathbb{R}^n$ is the state with x_0 unknown, $u_k \in \mathbb{R}^m$ is the control input, $y_k \in \mathbb{R}^p$ is the measured output, $v_k \in \mathbb{R}^n$ is unknown process noise, $w_k \in \mathbb{R}^p$ is unknown measurement noise, and A, B, and F are constant known matrices of appropriate dimensions. For the problem to be well posed, assume that the matrix pairs (A, B) and (F, A) are, respectively, reachable and observable. Suppose that the output sensors communicate with the controller over a digital channel that can carry one symbol s_k from a finite, possibly time-varying alphabet $\mathbb{S}_k$ of cardinality $M_k \geq 1$ during the $(k + 1)$-th sampling interval. Assume for simplicity that the channel is errorless, with negligible propagation delay. The asymptotic average rate at which the channel transports data may then be defined as

$$R := \varliminf_{k \to \infty} \frac{1}{k} \sum_{j=0}^{k-1} \log_2 M_j \quad \text{(bits/sample)}. \quad (6)$$

Note that if the channel alphabet $\mathcal{S}_k$ is constant or varies periodically with k, the inferior limit reduces to a straight limit.

In full generality, each transmitted symbol may depend on all past and present measurements and past symbols,

$$s_k = \gamma_k(y_0^k, s_0^{k-1}) \in \mathcal{S}_k, \quad \forall k \in \mathbb{Z}_{\geq 0}, \quad (7)$$

where γ_k is the coder mapping at time k. At time k the controller has $s_0, \ldots, s_k$ available and then applies a control law of the general form

$$u_k = \delta_k(s_0^k) \in \mathbb{R}^m, \quad \forall k \in \mathbb{Z}_{\geq 0}, \quad (8)$$

where δ_k is the controller mapping at time k.

In practice, additional memory or structural constraints are usually placed on the general coding and control rules (7) and (8). For instance, if a static quantizer of the form (1) is used, then the coding alphabet $\mathcal{S}_k \equiv \mathcal{S}$ will be constant and $s_k \equiv \gamma(y_k)$ will represent the index of the quantizer cell that contains y_k. Similarly, a static, memoryless controller is captured by setting $u_k = \delta(s_k)$ in (8).

Finite-dimensional coding and control laws may be formulated by defining internal coder and controller states ψ_k^γ and ψ_k^δ with local updates of the form

$$s_k = \gamma\left(y_k, \psi_{k-1}^\gamma\right), \quad \psi_k^\gamma = \phi\left(s_k, \psi_{k-1}^\gamma\right), \quad (9)$$

$$\psi_k^\delta = \eta\left(s_k, \psi_{k-1}^\delta\right), \quad u_k = \delta\left(\psi_k^\delta\right). \quad (10)$$

If the states ψ_k^γ and ψ_k^δ are finite valued, then the coding and control laws are called *finite-state*.

Additive Noise Model

Early approaches to quantized control modeled quantization errors as additive noise, in order to allow the use of well-developed tools from linear stochastic control (Curry 1970). While this was reasonable at high quantizer resolution, it failed to capture two key properties.

A simple example illustrates this. Consider a scalar, noiseless, fully observed, unstable LTI plant – i.e., (5) with $n = 1$, $A = a$ with $|a| > 1$, B, $C = 1$, and w_k, $v_k = 0$ – where x_0 is a random variable. Under static, high-resolution uniform quantization, the data available to the controller is expressed as a noisy linear measurement

$$y_k' := Q(x_k) = x_k + n_k, \quad k \in \mathbb{Z}_{\geq 0},$$

where the quantizer error process (n_k) is treated as zero mean white noise uncorrelated with (x_k) and having constant variance given by (3).

The first shortcoming of this approach is that it precludes the possibility of asymptotic mean-square stability, which would effectively require the controller to estimate the initial state x_0 with a mean-square error diminishing strictly faster than a^{-2k}. This turns out to be impossible under the uncorrelatedness assumption and the constraint $|a| > 1$.

However, in the seminal paper (Delchamps 1990), it was shown that asymptotic stability could in fact be achieved, by using a nonlinear controller that exploited the correlation between successive quantizer errors. To see this, suppose

that the unknown initial state x_0 is confined to a known interval $[-l_0, l_0]$. At time $k \geq 0$, suppose that $l_k \geq 0$, $k = 1, 2, \ldots$ represent bounds to be determined on the future states x_k. Let Q be a static one-bit quantizer – i.e., with $M = 2$ – such that $Q(x) = 1$ if $x \geq 0$ and $Q(x) = -1$ if $x < 0$. At time k let $u_k = -0.5al_k Q(x)$ so that

$$x_{k+1} = \begin{cases} a(x_k - 0.5l_k) & \text{if } 0 \leq x_k \leq x_k \\ a(x_k + 0.5l_k) & \text{if } -l_k \leq x_k < 0 \end{cases}.$$

$$\Rightarrow |x_{k+1}| \leq 0.5|a|l_k =: l_{k+1}.$$

If $|a| < 2$ then $l_k \to 0$, and asymptotic stability is achieved uniformly and with exponential convergence.

However, the main drawback of the additive white noise model is that it does not predict the loss of closed-loop stability that can result when the quantizer resolution is too coarse. This is because the number M of quantizer points only serves to determine the variance of the additive noise n_k: reducing M increases the variance of n_k and the mean-square states, but they remain bounded over time. In contrast, a rigorous analysis reveals that stability is impossible by any means, linear or nonlinear, when M drops below a certain threshold.

Numerous proofs of this loss of stability exist. In a stochastic setting, the argument is based on fixing the coder and controller and expanding out the closed-loop dynamics of the scalar LTI plant to write

$$x_k = a^k x_0 - a^k z_k \qquad (11)$$

where $z_k := -a^{-k} \sum_{j=0}^{k-1} a^{k-j-1} u_j$. As z_k is a function of $s_0^{k-1} \in \mathcal{S}^k$, it can take at most M^k values. Furthermore, in the absence of noise, it is fully determined by x_0, for a given coding and control policy (7) and (8). Thus z_k can be regarded as the output $Q'_k(x_0)$ of an M^k-valued quantizer. Substituting this into (11) yields

$$x_k = a^k(x_0 - Q'_k(x_0)).$$

From the asymptotic quantizer result (4), it then follows that for large k,

$$\mathrm{E}[x_k^2] \geq c \frac{a^{2k}}{M^{2k}} \left(\int f_{X_0}(x)^{1/3} \mathrm{d}\mu(x) \right)^3.$$

Thus a necessary condition for asymptotic mean-square stabilizability is that $M > |a|$ – see Nair and Evans (2000) for details.

The Data Rate Theorem

The discussions above emphasized the need for a more rigorous approach to quantized control. In the literature, the necessary condition $M > |a|$ was first derived in a nonrandom setting, where it was shown to be both sufficient and necessary to be able to ensure uniform stability (Baillieul 1999; Wong and Brockett 1999).

The sufficiency argument is constructive. Let Q be an M-level uniform quantizer on $[-1, 1]$, with cells formed by partitioning $[-1, 1]$ into M subintervals $R^1, \ldots, R^M$ of equal length and setting $Q(z)$ to be the midpoint of R^i when $z \in R^i$. Suppose that at time k the unknown state x_k lies in a known interval $[-l, l]$, and set $u_k = -alQ(x_k/l)$. Thus

$$|x_{k+1}| = |a||x_k - lQ(x_k/l)|$$
$$= |a|l \left| \frac{x}{l} - Q\left(\frac{x}{l}\right) \right| \leq |a|\frac{l}{M}.$$

When $M > |a|$, the right-hand side $< l$. Thus $x_{k+1} \in [-l, l]$ as well, and boundedness is achieved. Uniform asymptotic stability can be achieved by replacing the constant parameter l in the argument above with a time-varying bound l_k, updated as $l_{k+1} = |a|l_k/M \to 0$.

The necessity argument is based on volume partitioning. The basic idea is to fix an arbitrary coding and control policy and let m_k be the Lebesgue measure of the set of values that x_k can take at time $k \in \mathbb{Z}_{\geq 0}$. After k time steps, the plant dynamics expand this uncertainty volume m_0 by a factor $|a|^k$. However, the coder effectively divides this region into M^k disjoint, exhaustive pieces, each of which is shifted by the controller. As Lebesgue measure is translation invariant, it then

follows that $m_k \geq |a|^k m_0 / M^k$. Consequently M must exceed $|a|$ if the closed loop is uniformly asymptotically stable.

The tight criterion $M > |a|$, or equivalently $R > \log_2 |a|$, was the first instance of the *data rate theorem*. Volume-partitioning arguments and Jordan canonical forms can be used to generalize it to LTI plants with vector-valued states, yielding the necessary and sufficient condition

$$R > \sum_{i:|\lambda_i| \geq 1} \log_2 |\lambda_i| =: H, \qquad (12)$$

where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A. This criterion is remarkably universal, having been shown to be tight for a variety of settings and objectives: e.g., for asymptotic r-th moment stabilizability with random, unbounded x_0 and no process or measurement noise (Nair and Evans 2003); uniform stabilizability with bounded x_0 and no process or measurement noise (Baillieul 2002); uniform stabilizability with bounded initial state, process, and measurement noise (Hespanha et al. 2002; Tatikonda and Mitter 2004); and mean-square stabilizability with random, unbounded initial state, process, and measurement noise (Nair and Evans 2004).

The deep nature of (12) becomes even clearer when it is noted that the right-hand side of (12) coincides with the intrinsic entropy generation rate H of the (open-loop) plant, in both the Kolmogorov-Sinai and topological senses; that is, it describes the growth rate of the number of distinguishable state trajectories. Thus the data rate theorem states that stability is possible iff the communication rate in the feedback loop exceeds the rate at which the plant generates uncertainty. This interpretation leads to the notion of *feedback entropy* (see cross-reference to article by C. Kawan).

Zooming Quantized Control

When the plant noise and initial state of the plant (5) are bounded, stability (in a uniform sense) can be guaranteed by applying a linear observer to track the plant states with bounded error and then applying a suitable static, memoryless coding and control policy on the observer states x_k^o.

However, if the noise or initial state has unbounded support – e.g., when they are Gaussian or when prior bounds on them are not known – then stability cannot be achieved by any such static memoryless scheme or indeed by any scheme where the control inputs (8) are bounded (Nair and Evans 2004). The explanation is simple: due to the infinite support, there is a nonzero probability that the propagated state Ax_t will be beyond reach of the control input at some time t. The unstable plant dynamics then amplify this shortfall, causing the same phenomenon to occur with increasing probability at subsequent times and inevitably leading to instability.

One solution is to use a *zooming quantizer*, i.e., having a dynamic range $l_k > 0$ that is not bounded a priori but expands or contracts according to the most recent symbol (Brockett and Liberzon 2000). In the noiseless case, if this symbol corresponds to the "overload region" of the quantizer (as indicated by a special symbol), then the range is updated as $l_{k+1} := \phi_{out} l_k$, where $\phi_{out} > 1$ is the "zoom-out" factor. Otherwise $l_{k+1} := \phi_{in} l_k$, where $\phi_{in} < 1$ is the "zoom-in" coefficient.

In the communications literature such schemes are called *adaptive quantizers* (Goodman and Gersho 1974). If ϕ_{out} is sufficiently large compared to the unstable open-loop eigenvalues, and if ϕ_{in} is not too small, then global asymptotic stability ensues. With unbounded noise in the plant, variants of this scheme guarantee mean-square stability at any data rate satisfying (12) (Nair and Evans 2004) or *input-to-state stability* (Liberzon and Nesic 2007).

Zooming quantization is an important example of a finite-dimensional coder-controller (9) and (10), with l_k playing the role of an internal state variable. As the range update is driven by the symbols, both coder and controller can each generate identical copies of l_k, provided that there are no errors in the channel and they both start from the same initial range l_0. The important issue of how to design a scheme that can cope with mismatched initial internal states or a small level of channel errors is as yet largely unexplored.

Erroneous Digital Channels

The information-theoretic aspects of quantized control become especially pronounced when the channel is not error-free. In this case, the data rate theorem (12) can be extended, but in ways that are highly dependent on the precise setting and stability objective.

A common figure of merit for a stochastic discrete memoryless channel (DMC) is its *ordinary capacity C*. This is defined operationally as the largest block-code bit rate that can be transmitted across the channel with negligible probability of decoding error, and also coincides with the largest rate of Shannon information across the channel (Shannon 1948). For a noiseless LTI plant with random initial state controlled over a DMC, the condition $C > H$ is a tight criterion for almost sure (a.s.) asymptotic stabilizability (Matveev and Savkin 2007a). This is a natural generalization of (12).

On the other hand, if the objective is to bound the state moments of a scalar LTI plant subject to bounded process noise, then the achievability of this goal is determined by the *anytime capacity C_{any}* (Sahai and Mitter 2006): this is essentially given by the fastest decay rate of the decoding error probability.

However, if the aim is a.s. boundedness of an LTI plant with random initial state and bounded, nonstochastic process noise, then the stabilizability criterion changes again to $C_{0f} > H$ (Matveev and Savkin 2007b). Here C_{0f} is the *zero-error feedback capacity* of the channel, defined as the largest block-code bit rate that can be transmitted across the channel with *exactly zero* probability of decoding error and with perfect channel feedback (Shannon 1956).

As $C_{0f} < C_{any} < C$ for most channels, these conditions do not coincide. This suggests that there is no universal, operationally relevant information theory for feedback control over error-prone channels: such a theory must instead be tailored to match the underlying objectives and assumptions. For systems with nonstochastic disturbances, preliminary steps in this direction have been taken in Nair (2012, 2013). The reader is also referred to You and Xie (2011) and Minero et al. (2013) for information-theoretic analyses of stochastic linear systems controlled via Markov channels.

Summary and Future Directions

This article described the key elements of quantized control with finite data rates, emphasizing the interplay between coding and control. A great deal is now known about the fundamental limitations on stability in quantized control systems consisting a single feedback loop. Two major directions for future research suggest themselves:

– Little work has been done on designing optimal coding and control schemes or determining optimal costs at a given rate, apart from one or two special cases and structural results – see Nair et al. (2007) and the references therein. It is very unlikely that explicit, closed-form solutions will be possible. However, numerical approaches based on the Lloyd-Max algorithm, particle filtering, and model-predictive control may prove fruitful.

– Networked control systems usually consist of a number of subsystems interconnected over a network. Furthermore, in multi-agent systems the main objective may not be stability, but rather coordination or consensus to a common state. Comparatively little is known about the data rate requirements and information-theoretic aspects of these problems.

Cross-References

▶ Data Rate of Nonlinear Control Systems and Feedback Entropy

Bibliography

Baillieul J (1999) Feedback designs for controlling device arrays with communication channel bandwidth constraints. In: ARO workshop on smart structures, Pennsylvania State University

Baillieul J (2002) Feedback designs in information-based control. In: Pasik-Duncan B (ed) Stochastic theory and control: proceedings of a workshop held in Lawrence, Kansas. Springer, pp 35–57

Brockett RW, Liberzon D (2000) Quantized feedback stabilization of linear systems. IEEE Trans Autom Control 45(7):1279–1289

Curry RE (1970) Estimation and control with quantized measurements. MIT, Cambridge

Delchamps DF (1990) Stabilizing a linear system with quantized state feedback. IEEE Trans Autom Control 35:916–924

Gersho A, Gray RM (1993) Vector quantization and signal compression. Kluwer, Boston

Goodman DJ, Gersho A (1974) Theory of an adaptive quantizer. IEEE Trans Comms 22: 1037–1045

Hespanha J, Ortega A, Vasudevan L (2002) Towards the control of linear systems with minimum bit-rate. In: Proceedings of the 15th international symposium on the mathematical theory of networks and systems (MTNS), U. Notre Dame

Liberzon D, Nesic D (2007) Input-to-state stabilization of linear systems with quantized state measurements. IEEE Trans Autom Control 52:767–781

Matveev AS, Savkin AV (2007a) An analogue of Shannon information theory for detection and stabilization via noisy discrete communication channels. SIAM J Control Optim 46(4): 1323–1367

Matveev AS, Savkin AV (2007b) Shannon zero error capacity in the problems of state estimation and stabilization via noisy communication channels. Int J Control 80:241–255

Minero P, Coviello L, Franceschetti M (2013) Stabilization over Markov feedback channels: the general case. IEEE Trans Autom Control 58(2):349–362

Nair GN (2012) A nonstochastic information theory for feedback. In: Proceedings of the IEEE conference decision and control, Maui, pp 1343–1348

Nair GN (2013) A nonstochastic information theory for communication and state estimation. IEEE Trans Autom Control 58(6):1497–1510

Nair GN, Evans RJ (2000) Stabilization with data-rate-limited feedback: tightest attainable bounds. Syst Control Lett 41(1):49–56

Nair GN, Evans RJ (2003) Exponential stabilisability of finite-dimensional linear systems with limited data rates. Automatica 39:585–593

Nair GN, Evans RJ (2004) Stabilizability of stochastic linear systems with finite feedback data rates. SIAM J Control Optim 43(2):413–436

Nair GN, Fagnani F, Zampieri S, Evans RJ (2007) Feedback control under data rate constraints: an overview. Proc IEEE 95(1):108–137. In special issue on Technology of Networked Control Systems

Sahai A, Mitter S (2006) The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link part 1: scalar systems. IEEE Trans Inf Theory 52(8): 3369–3395

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423, 623–656. Reprinted in 'Claude Elwood Shannon Collected Papers', IEEE Press, 1993

Shannon CE (1956) The zero-error capacity of a noisy channel. IRE Trans Inf Theory 2:8–19

Tatikonda S, Mitter S (2004) Control under communication constraints. IEEE Trans Autom Control 49(7):1056–1068

Wong WS, Brockett RW (1999) Systems with finite communication bandwidth constraints II: stabilization with limited information feedback. IEEE Trans Autom Control 44:1049–1053

You K, Xie L (2011) Minimum data rate for mean square stabilizability of linear systems with Markovian packet losses. IEEE Trans Autom Contr 56(4): 772–785

Q

R

Randomized Methods for Control of Uncertain Systems

Fabrizio Dabbene and Roberto Tempo
CNR-IEIIT, Politecnico di Torino, Torino, Italy

Abstract

In this article, we study the tools and methodologies for the analysis and design of control systems in the presence of random uncertainty. For analysis, the methods are largely based on the Monte Carlo simulation approach, while for design new randomized algorithms have been developed. These methods have been successfully employed in various application areas, which include systems biology; aerospace control; control of hard disk drives; high-speed networks; quantized, embedded, and electric circuits; structural design; and automotive and driver assistance.

Keywords

Chernoff bound; Hoeffding inequality; Monte Carlo simulation; Randomization algorithms

Preliminaries

Randomized methods for control deal with the design of uncertain and complex systems. They have been originally developed for linear systems affected by structured uncertainty, usually expressed in the so-called $M - \Delta$ configuration. A similar approach may be followed when dealing with uncertainty in other contexts, such as uncertainty in the environment (random disturbances) or even when there is no uncertainty in the problem formulation, but the complexity of the problem is such that randomized methods may be the best approach, since these methods are known to break the curse of dimensionality, see Tempo et al. (2013) for details.

For the sake of simplicity, we consider here an uncertain plant transfer function $P(s, q)$ affected by parametric uncertainty

$$q = [q_1 \ldots q_\ell]^T$$

bounded in a set $\mathbb{Q} \subset \mathbb{R}^\ell$. The objective is to design the parameters $\theta \in \mathbb{R}^n$ of a controller transfer function $C(s, \theta)$ so to guarantee robustly some desired performance. This is reformulated as the problem of finding a design satisfying some *uncertain constraints* of the form

$$f(\theta, q) \leq \text{ for all } q \in \mathbb{Q}.$$

In other words, the goal is to design a robust controller which satisfies the uncertain constraints. Specific examples of these constraints include an $\mathcal{H}_\infty$ or $\mathcal{H}_2$ norm bound on the closed-loop sensitivity function or time-domain specifications.

Since this objective may be too hard to achieve in many situations, we are relaxing it as follows:

we would like to design controller parameters $\theta \in \mathbb{R}^n$ such that a certain *violation* is allowed, i.e.,

$$\begin{cases} f(\theta, q) \leq 0 \text{ for all } q \in \mathbb{Q}_{\text{good}}; \\ f(\theta, q) > 0 \text{ for all } q \in \mathbb{Q}_{\text{bad}} \end{cases}$$

where the good and bad sets satisfy the equations

$$\begin{cases} \mathbb{Q}_{\text{good}} \cup \mathbb{Q}_{\text{bad}} = \mathbb{Q}; \\ \mathbb{Q}_{\text{good}} \cap \mathbb{Q}_{\text{bad}} = \varnothing, \end{cases}$$

and the goal is to guarantee that the bad set $\mathbb{Q}_{\text{bad}}$ is "small" enough. To state this concept more precisely, we assume that $q \in \mathbb{Q}$ is a random vector with given probability density function (pdf), and we introduce the probability of violation and the controller reliability.

Definition 1 (Probability of Violation and Reliability) The probability of violation for the controller parameters $\theta \in \mathbb{R}^n$ is defined as

$$V(\theta) \doteq \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\}.$$

The reliability of the design $\theta \in \mathbb{R}^n$ is given by

$$R(\theta) = 1 - V(\theta).$$

In this context, we are satisfied if, given a violation level $\propto \in (0, 1)$, the probability of violation is sufficiently small, i.e., $V(\theta) \leq \propto$. We remark that relaxing the requirement of robust satisfaction of the uncertain constraints $f(\theta, q) \leq 0$ to a probabilistic one (by means of the probability of violation) is not helpful computationally because computing *exactly* the probability $V(\theta)$ is very hard in general because it requires to solve a multidimensional integral over the nonconvex domain defined by $f(\theta, q) > 0$, with $q \in \mathbb{Q} \subset \mathbb{R}^\ell$. The problem is then resolved introducing Monte Carlo randomized algorithms (formally defined in the next section). This is a computational approach which leads to solutions which are often denoted as PAC (probably approximately correct) (Vidyasagar 2002).

More precisely, for fixed design $\theta \in \mathbb{R}^n$, to compute a Monte Carlo approximation based on N random simulations, we generate N independent identically distributed (iid) random

samples of $q \in \mathbb{Q}$, called the *multisample*, of the uncertainty q according to the given probability density function

$$q^{(1...N)} = \left\{ q^{(1)}, \ldots, q^{(N)} \right\} \in \mathbb{Q}^N.$$

The cardinality N of the multisample $q^{(1...N)}$ is often referred to as the *sample complexity* (Vidyasagar 2001). The empirical violation of the design θ is then defined.

Definition 2 (Empirical Violation) *For given* $\theta \in \mathbb{R}^n$, *the empirical violation of* $V(\theta, q)$ *with respect to the multisample* $q^{(1...N)} = \left\{ q^{(1)}, \ldots, q^{(N)} \right\} \in \mathbb{Q}^N$ *is given by*

$$\hat{V}_N\left(\theta, q^{(1...N)}\right) \doteq \frac{1}{N} \sum_{i=1}^N \mathbb{I}_f\left(\theta, q^{(i)}\right)$$

where $\mathbb{I}_f\left(\theta, q^{(i)}\right)$ is the indicator function

$$\mathbb{I}_f\left(\theta, q^{(i)}\right) \doteq \begin{cases} 0 \text{ if } f\left(\theta, q^{(i)}\right) \leq 0 \\ 1 \qquad \text{otherwise.} \end{cases}$$

Monte Carlo Randomized Algorithms for Analysis

In this section, we study Monte Carlo randomized algorithms for analysis, i.e., when the controller parameters are fixed, and in particular we concentrate on a PAC computation of the probability of violation. In agreement with classical notions in computer science (Mitzenmacher and Upfal 2005; Motwani and Raghavan 1995), a randomized algorithm (RA) is formally defined as an algorithm that *makes random choices* during its execution to produce a result. This implies that, even for the same input data, the algorithm might produce different results at different runs, and, moreover, the results may be incorrect. Therefore, statements regarding properties of these algorithms are necessarily of probabilistic nature.

Formally, the probabilistic parameters ε, $\delta \in (0, 1)$ called *accuracy* and *confidence,* respectively, are introduced. For any θ, the PAC approach provides an empirical violation which is

an approximation $\hat{V}_N\left(\theta, q^{(1...N)}\right)$ to $V(\theta)$ within accuracy ε, and this event holds with confidence $1 - \delta$.

Monte Carlo Randomized Algorithm

Given a design $\theta \in \mathbb{R}^n$, a Monte Carlo randomized algorithm (MCRA) is a randomized algorithm that provides an approximation $\hat{V}_N\left(\theta, q^{(1...N)}\right)$ to $V(\theta)$ based on the multisample $q^{(1...N)}$. Given accuracy ε and confidence δ, the approximation may be incorrect, i.e.,

$$\left| V(\theta) - \hat{V}\left(\theta, q^{(1...N)}\right) \right| > \epsilon$$

but the probability of such an event is bounded, and it is smaller than δ.

In general, the results obtained by an MCRA as well as its running time would be different from one run to another since the algorithm is based on random sampling. As a consequence, the computational complexity of such an algorithm is usually measured in terms of its expected running times. MCRA are efficient because the expected running time is of polynomial order in the problem size (Tempo et al. 2013). One-sided and two-sided Monte Carlo randomized algorithms may be also defined (Tempo and Ishii 2007).

To derive the probabilistic properties of MCRA, we need to state the so-called Hoeffding inequality, which provides a bound on the error between the probability of violation and the empirical violation (Vidyasagar 2002).

Two-Sided Hoeffding Inequality

For fixed $\theta \in \mathbb{R}^n$ and $\varepsilon \in (0, 1)$, we have

$$\text{Prob}\left\{ q^{(1...N)} \in \mathbb{Q}^N : \left| V(\theta) - \hat{V}\left(\theta, q^{(1...N)}\right) \right| \right.$$
$$\left. > \epsilon \right\} \leq 2e^{-2N\epsilon^2}.$$

For fixed accuracy ε, we observe that the right-hand side of this equation approaches zero exponentially. Furthermore, if we bound the right-hand side of this equation with confidence δ, we immediately obtain the classical (additive)

Chernoff bound (Chernoff 1952) which is stated next.

Chernoff Bound

For any $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$, if

$$N \geq \frac{1}{2\epsilon^2} \log \frac{2}{\delta}$$

then, with probability greater than $1 - \delta$, we have

$$\left| V(\theta) - \hat{V}\left(\theta, q^{(1...N)}\right) \right| \leq \epsilon.$$

The Chernoff bound provides an indication of the required sample size, i.e., it provides the so-called sample complexity. More precisely, the sample complexity of a randomized algorithm is defined as the minimum cardinality of the multisample $q^{(1...N)}$ that needs to be drawn in order to achieve the desired accuracy ε and confidence δ. Notice that the confidence enters the Chernoff bound in a logarithmic fashion, while accuracy enters quadratically, and therefore, it is much more expensive computationally. Other large deviation inequalities and sample complexity bounds are discussed in the literature, including in particular the (multiplicative) Chernoff bound and the log-over-log bound for computing the so-called empirical maximum (Tempo et al. 1997). We refer to Vidyasagar (2002) for additional details.

Remark 1 (Las Vegas Randomized Algorithms) Las Vegas randomized algorithms (LVRA) are based on random samples generated according to a *discrete* probability density function, instead of a continuous pdf as in the case of Monte Carlo. Therefore, contrary to MCRA, LVRA provide the "correct answer" with probability one because the entire search space can be fully explored. However, because of randomization, the running time of an LVRA is random (similarly to MCRA) and may be different in each execution. Hence, it is of interest to study the expected running time of the algorithm. It is noted that the expectation is with respect to the random samples generated during the execution of the algorithm and not to the problem data. Classical examples of LVRA are within computer science

and include the well-known randomized quick-sort (RQS) algorithm for ranking numbers, which is implemented in a C library of the UNIX operating system (Knuth 1998). Other more recent developments in systems and control regarding these algorithms are for the PageRank computation in the Google search engine (Ishii and Tempo 2010), consensus over large-scale networks (Fagnani and Zampieri 2008), localization and coverage control of robotic networks (Bullo et al. 2012), and opinion dynamics (Frasca et al. 2013). These problems are generally formulated in a graph theoretic setting consisting of nodes and links, and either the nodes or the links are randomly selected according to a given "local" protocol (often called gossip) based on a given discrete pdf.

Randomized Algorithms for Control Design

This section deals with control problems which require computing a design $\theta \in \mathbb{R}^n$ satisfying some probabilistic properties on the uncertain constraints $f(\theta, q)$. Two classes of problems, feasibility and optimization, are considered.

Feasibility Problem

Given uncertain constraints $f(\theta, q)$ and level $\propto \in (0, 1)$, compute $\theta \in \mathbb{R}^n$ such that

$$V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\} \leq \propto . \quad (1)$$

The second problem relates to the optimization of a linear function of the design parameters under probability constraints.

Optimization Problem

Given uncertain constraints $f(\theta, q)$, a linear objective function $c^T \theta$ and level $p \in (0, 1)$, solve the constrained optimization problem

$$\min_\theta \quad c^T \theta$$

$$\text{subject to } V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\}$$

$$\leq p. \quad (2)$$

Optimization problems subject to constraints of the form $V(\theta) = \text{Prob}\{q \in \mathbb{Q} : f(\theta, q) > 0\} \leq \propto$ are often called chance constraint optimization (Uryasev 2000).

Most of the algorithms that have been studied in the literature follow two main paradigms and are often based on the following convexity assumption.

Convexity Assumption

The uncertain constraint $f(\theta, q)$ is convex in θ for any fixed value of $q \in \mathbb{Q}$.

The two solution paradigms that have been proposed are now summarized. The algorithms have been implemented in the Toolbox RACT (Randomized Algorithms Control Toolbox) for probabilistic analysis and control design in the presence of uncertainty (Tremba et al. 2008).

Paradigm 1 (Sequential Approach)

Under the convexity assumption, we study the Feasibility Problem (1). The algorithms presented in the literature (see, e.g., (Calafiore et al. 2011) for finding a probabilistic feasible design) follow a general iterative scheme (Fig. 1), which consists of successive randomization steps to handle uncertainty and optimization steps to update the design parameters. In particular, these algorithms share two fundamental ingredients:

1. A *probabilistic oracle* which performs a random check, with the objective to assess whether the probability of violation $V(\theta^{(k)})$ of the current candidate solution $\hat{\theta}^{(k)}$ is smaller than a given level p and returns a *certificate* of unfeasibility, that is, a value $q^{(k)}$ such that $f(\hat{\theta}^{(k)}, q^{(k)}) > 0$, when the candidate solution is found unfeasible

2. An *update rule* ψ_{upd} which exploits the convexity of the problem for constructing a new candidate solution $\hat{\theta}^{(k+1)}$ based on the probabilistic oracle outcome

In this paradigm, the algorithm returns a design $\hat{\theta}_k$ such that

$$V\left(\hat{\theta}_k\right) = \text{Prob}\left\{q \in \mathbb{Q} : f\left(\hat{\theta}_k, q\right) > 0\right\} \leq \rho$$
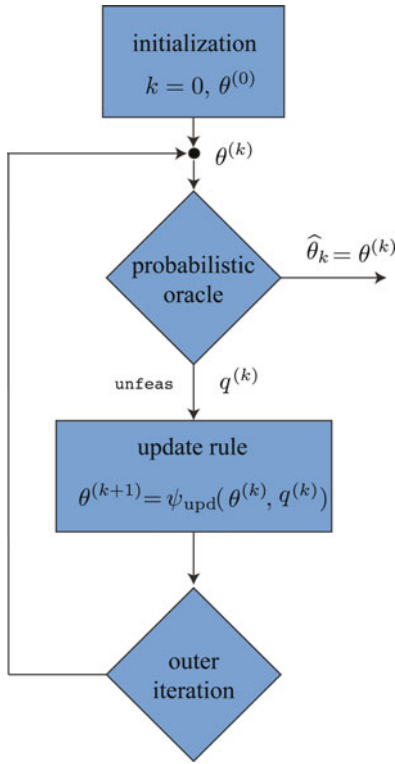
Randomized Methods for Control of Uncertain Systems, Fig. 1 Paradigm for sequential design consisting of probabilistic oracle and update rule

In this paradigm, the algorithm returns in one-shot a design $\hat{\theta}_N$ and the sample complexity N such that

$$V\left(\hat{\theta}_N\right) = \mathrm{Prob}\left\{q \in \mathbb{Q} : f\left(\hat{\theta}_N, q\right) > 0\right\} \leq \rho$$

is larger than $1-\delta$. That is, the violation probability associated to the design $\hat{\theta}_N$ is smaller than the level p, and this event holds with large confidence $1 - \delta$.

Concluding Remarks

Other probabilistic approaches have been proposed in the literature for control design, which are not based on the convexity assumption. A noticeable example is the strategy based on statistical learning theory (Valiant 1984; Vapnik 1998) which has the objective to design a controller without any convexity assumptions (Alamo et al. 2009). In particular, in Alamo et al. (2013), the general class of sequential probabilistic validation (SPV) algorithms has been introduced. A specific SPV algorithm tailored to scenario problems, providing a sequential scheme for dealing with the optimization problem, has been recently studied in Chamanbaz et al. (2013).

Cross-References

is larger than $1-\delta$. That is, the violation probability associated to the design $\hat{\theta}_k$ is smaller than the level ρ, and this event holds with large confidence $1 - \delta$.

Paradigm 2 (Scenario Approach)
Under the convexity assumption, we study the optimization problem (2). We remark that, even under these assumptions, solving this problem is very hard computationally because the probabilistic constraint is nonconvex. To alleviate this difficulty, we reformulate problem (2) as a so-called scenario problem introduced in Calafiore and Campi (2006), which is now described.

For randomly extracted scenarios $q^{(1\ldots N)}$, this approach requires to compute $\theta \in \mathbb{R}^n$ that solves the convex optimization problem subject to a finite number of sampled constraints

$$\hat{\theta}_N = \begin{array}{ll} \min_\theta & c^T \theta \\ \text{subject to} & f\left(\theta, q^i\right) \leq 0, \ i = 1, \ldots, N \end{array} \tag{3}$$

Bibliography

Alamo T, Tempo R, Camacho E (2009) A randomized strategy for probabilistic solutions of uncertain feasibility and optimization problems. IEEE Trans Autom Control 54:2545–2559

Alamo T, Tempo R, Luque A, Ramirez D (2013) The sample complexity of randomized methods for analysis and design of uncertain systems. arXiv:13040678 (accepted for publication)

Bullo F, Carli R, Frasca P (2012) Gossip coverage control for robotic networks: dynamical systems on the space of partitions. SIAM J Control Optim 50(1):419–447

Calafiore G, Campi M (2006) The scenario approach to robust control design. IEEE Trans Autom Control 51(1):742–753

Calafiore G, Dabbene F, Tempo R (2011) Research on probabilistic methods for control system design. Automatica 47:1279–1293

Chamanbaz M, Dabbene F, Tempo R, Venkataramanan V, Wang Q (2013) Sequential randomized algorithms for convex optimization in the presence of uncertainty. arXiv: 1304.2222

Chernoff H (1952) A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. Ann Math Stat 23:493–507

Fagnani F, Zampieri S (2008) Randomized consensus algorithms over large scale networks. IEEE J Sel Areas Commun 26(4):634–649

Frasca P, Ravazzi C, Tempo R, Ishii H (2013) Gossips and prejudices: ergodic randomized dynamics in social networks. In: Proceedings of the 4th IFAC workshop on distributed estimation and control in networked systems, Koblenz

Ishii H, Tempo R (2010) Distributed randomized algorithms for the PageRank computation. IEEE Trans Autom Control 55:1987–2002

Knuth D (1998) The art of computer programming. Sorting and searching, vol 3. Addison-Wesley, Reading

Mitzenmacher M, Upfal E (2005) Probability and computing: randomized algorithms and probabilistic analysis. Cambridge University Press, Cambridge

Motwani R, Raghavan P (1995) Randomized algorithms. Cambridge University Press, Cambridge

Tempo R, Ishii H (2007) Monte Carlo and Las Vegas randomized algorithms for systems and control: an introduction. Eur J Control 13:189–203

Tempo R, Bai EW, Dabbene F (1997) Probabilistic robustness analysis: explicit bounds for the minimum number of samples. Syst Control Lett 30:237–242

Tempo R, Calafiore G, Dabbene F (2013) Randomized algorithms for analysis and control of uncertain systems, with applications. Communications and control engineering series, 2nd edn. Springer, London

Tremba A, Calafiore G, Dabbene F, Gryazina E, Polyak B, Shcherbakov P, Tempo R (2008) RACT: randomized algorithms control toolbox for MATLAB. In: Proceedings 17th IFAC world congress, Seoul, pp 390–395

Uryasev SP (ed) (2000) Probabilistic constrained optimization: methodology and applications. Kluwer Academic, New York

Valiant L (1984) A theory of the learnable. Commun ACM 27(11):1134–1142

Vapnik V (1998) Statistical learning theory. Wiley, New York

Vidyasagar M (2001) Randomized algorithms for robust controller synthesis using statistical learning theory. Automatica 37:1515–1528

Vidyasagar M (2002) Learning and generalization: with applications to neural networks, 2nd edn. Springer, New York

Realizations in Linear Systems Theory

Panos J. Antsaklis[1] and A. Astolfi[2,3]
[1]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
[2]Department of Electrical and Electronic Engineering, Imperial College London, London, UK
[3]Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

Abstract

When a state variable description of a linear system is known, then its input–output behavior can be easily realized by interconnecting simpler components. The problem of realization refers to the following: given an input–output description such as the impulse response, or the transfer function in the case of time-invariant systems, find a state variable description, the impulse response of which is the given one. Existence and minimality conditions are discussed. We are interested in realizations of minimum order which is the case when the realization is both controllable and observable. Realizations in both the continuous-time and discrete-time systems are discussed.

Keywords

Controllability; Irreducible; Minimal order; Observability; Realizations

Introduction

The problem of system realization is as follows: given an external description of a linear system, specifically its impulse response (or its transfer function in the case of a time-invariant system), determine an internal state variable description that generates the given impulse response (or the

transfer function). The name reflects the fact that if a (continuous-time) state variable description is known, an operational amplifier circuit can be easily built to realize (actually simulate) the system response.

Before we discuss realizations, we review the key relations between internal state variable and external impulse response or transfer function descriptions.

Consider a system described by

$$\dot{x} = A(t)x + B(t)u, \quad y = C(t)x + D(t)u, \tag{1}$$

where $x(t)$, the state vector, is a column vector of dimension n $(x(t) \in \mathbb{R}^n)$ and $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$ are the inputs and outputs of the system. $A(t) \in \mathbb{R}^{n \times n}$, $B(t) \in \mathbb{R}^{n \times m}$, $C(t) \in \mathbb{R}^{p \times n}$, $D(t) \in \mathbb{R}^{p \times m}$ with entries continuous functions. The output response is given by

$$y(t) = C(t)\Phi(t, t_0)x_0 + \int_0^t H(t, \tau)u(\tau)d\tau, \tag{2}$$

where $\Phi(t, t_0)$ is the $n \times n$ transition matrix of $\dot{x} = A(t)x$, $x(t_0) = x_0$ is the initial condition, and $H(t, \tau)$ is the $p \times m$ impulse response matrix given by

$$H(t, \tau) = \begin{cases} C(t)\Phi(t, \tau)B(t) \\ \quad + D(t)\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau, \end{cases} \tag{3}$$

where $\delta(t - \tau)$ is the impulse (delta or Dirac) function applied at time $t = \tau$. Recall that $H(t, \tau)$ denotes the response at time t when an impulse input is applied at time τ assuming zero initial conditions.

In the time-invariant system, (1) becomes

$$\dot{x} = Ax + Bu, \quad y = Cx + Du, \tag{4}$$

and the output response in this case is

$$y(t) = Ce^{At}x_0 + \int_0^t H(t, \tau)u(\tau)d\tau, \tag{5}$$

where, without loss of generality, the initial time t_0 was taken to be zero. In this case, the impulse response is

$$H(t, \tau) = \begin{cases} Ce^{A(t-\tau)}B + D\delta(t - \tau) & t \geq \tau, \\ 0 & t < \tau. \end{cases} \tag{6}$$

Recall that time invariance implies that $H(t, \tau) = H(t - \tau, 0)$, and so τ, which is the time an impulse input is applied to the system, can be taken to be zero $(\tau = 0)$, without loss of generality, to give $H(t, 0)$. The transfer function of the system is the (one-sided or unilateral) Laplace transform of $H(t, 0)$, namely,

$$H(s) = \mathcal{L}[H(t, 0)] = C(sI - A)^{-1}B + D. \tag{7}$$

A realization of $H(t, \tau)$ is any state variable description (1), $\{A(t), B(t), C(t), D(t)\}$, the impulse response of which is $H(t, \tau)$, that is, (3) is satisfied, similarly for the time-invariant case when (6) is satisfied.

In the time-invariant case, a realization is commonly defined in terms of the transfer function matrix $H(s)$. Then a realization of $H(s)$ is any state variable description (4), $\{A, B, C, D\}$, the transfer function of which is $H(s)$, that is, (7) is satisfied.

There are alternative conditions under which a set of $\{A, B, C, D\}$ is a realization of some $H(s)$. To this end, expand $H(s)$ in a Laurent series to obtain

$$H(s) = H_0 + H_1 s^{-1} + H_2 s^{-2} + \cdots. \tag{8}$$

The matrices H_i, $i = 0, 1, 2, \ldots$ are called *Markov parameters* of the system and can be determined as follows:

$$H_0 = \lim_{s \to \infty} H(s), \quad H_1 = \lim_{s \to \infty} s(H(s) - H_0),$$

$$H_k = \lim_{s \to \infty} s^k (H(s) - \Sigma_{i=0}^{k-1}H_i s^{-i}), \quad k \geq 1.$$

It can be shown that a set $\{A, B, C, D\}$ is a realization of $H(s)$ if and only if

$$H_0 = D \text{ and } H_i = CA^{i-1}B, \quad i = 1, 2, \ldots. \tag{9}$$

Below we introduce conditions for the existence of a realization given $H(t, \tau)$ or $H(s)$.

Note that if a realization does exist, then there is an infinite number of realizations. One could see this, for example, by considering equivalent descriptions of a realization – all have the same impulse response or transfer function.

Existence and Minimality

It can be shown that $H(t, \tau)$ is realizable as the impulse response of a system described by (1) if and only if $H(t, \tau)$ can be decomposed into

$$H(t, \tau) = M(t)N(\tau) + D(t)\delta(t - \tau), \quad (10)$$

for $t \geq \tau$, where M, N, and D are $p \times n$, $n \times m$, and $p \times m$ matrices, respectively, with continuous real-valued entries and with n finite. If in addition to (10), $M(t)$ and $N(t)$ are differentiable and

$$H(t, \tau) = H(t - \tau, 0), \quad (11)$$

then $H(t, \tau)$ is realizable as the impulse response of a system described by a time-invariant system (4).

In the time-invariant case, it is more common to work with a given transfer function $H(s)$. Then $H(s)$ is realizable, as the transfer function matrix of a time-invariant system described by (4), if and only if $H(s)$ is a matrix of rational functions and satisfies

$$\lim_{s \to \infty} H(s) < \infty, \quad (12)$$

that is, if and only if $H(s)$ is a proper rational matrix or equivalently if and only if

$$\lim_{s \to \infty} H(s) = D \quad (13)$$

is a constant.

We are interested in realizations (4) of a given transfer function matrix $H(s)$ of least order n ($A \in \mathbb{R}^{n \times n}$), called minimal or irreducible realizations of $H(s)$.

The following two results completely solve the minimal realization problem.

Theorem 1 *An n-dimensional realization $\{A, B, C, D\}$ of $H(s)$ is minimal (irreducible, of least order) if and only if it is both reachable (or controllable) and observable.*

Note that if (A, B) is not controllable, then by separating the controllable and uncontrollable parts of the system by an equivalence transformation and taking only the controllable part, one can still obtain $H(s)$ because the uncontrollable part of the system cancels out in $H(s)$. Similarly for observability. So controllability and observability are necessary conditions for minimality. It can be shown that they are also sufficient.

Theorem 2 *If a minimal realization of order n is found, then any other minimal realization may be obtained via equivalence transformation.*

Specifically, if $\{A, B, C, D\}$ and $\{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ are realizations of $H(s)$ and $\{A, B, C, D\}$ is minimal, then $\{\bar{A}, \bar{B}, \bar{C}, \bar{D}\}$ is also minimal if and only if there exists a nonsingular matrix P such that

$$\bar{A} = PAP^{-1}, \ \bar{B} = PB, \ \bar{C} = CP^{-1}, \ \bar{D} = D. \quad (14)$$

Discrete-Time Linear Systems

The definitions and results for the discrete-time case are completely analogous to the ones in the continuous-time case. They are summarized below for completeness.

Consider systems described by

$$\begin{aligned} x(k + 1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k). \end{aligned} \quad (15)$$

The output response is

$$y(k) = C(k)\Phi(k, k_0)x_0 + \sum_{i=k_0}^{k-1} H(k, i)u(i), \ k > k_0, \quad (16)$$

where $\Phi(k, k_0)$ is the $n \times n$ transition matrix and $H(k, i)$ is the $p \times m$ discrete-impulse (pulse) response:

$$\Phi(k,l) = \begin{cases} A(k-1)\cdots A(l) & k > l, \\ I & k = l, \end{cases} \quad (17)$$

$$H(k,i) = \begin{cases} C(k)\Phi(k,i+1)B(i) & k > i, \\ D(k) & k = i, \\ 0 & k < i. \end{cases}$$
$$(18)$$

In the time-invariant case, Eqs. (15) and (16) become

$$x(k+1) = Ax(k) + Bu(k),$$
$$y(k) = Cx(k) + Du(k), \quad (19)$$

and

$$y(k) = CA^k x_0 + \sum_{i=0}^{k-1} H(k,i)u(i), \quad k > 0,$$
$$(20)$$

where, without loss of generality, k_0 is taken to be zero.

The discrete-impulse (pulse) response is now given by

$$H(k,i) = \begin{cases} CA^{k-(i+1)}B & k > i, \\ D & k = i, \\ 0 & k < i. \end{cases} \quad (21)$$

Since the system is time invariant, $H(k,i) = H(k-i,0)$ and i, the time the pulse input is applied, can be taken to be zero. The transfer function matrix for (19) is the (one-sided or unilateral) z-transform of $H(k,0)$:

$$H(z) = \mathcal{Z}\{H(k,0)\} = C(zI-A)^{-1}B + D. \quad (22)$$

It can be shown that given a $p \times m$ matrix $H(k,i)$, $k \geq i$, it is realizable as the pulse response of a system (15) if and only if $H(k,i)$ can be decomposed as

$$H(k,i) = \begin{cases} M(k)N(i) & k > i, \\ D(k) & k = i. \end{cases} \quad (23)$$

If, in addition, $H(k,i) = H(k-i,0)$, then it is realizable via a time-invariant description (19).

Similarly to the continuous-time case, $H(z)$ is realizable as the transfer function matrix of a system described by (19) if and only if

$$\lim_{z \to \infty} H(z) < \infty. \quad (24)$$

A realization (19) of $H(z)$ is minimal if and only if it is reachable (controllable from the origin) and observable. And if (19) is a minimal realization of $H(z)$, then any other minimal realization is equivalent to (19).

Realization Algorithms

Given a transfer function $H(s)$ (or $H(z)$), we are interested in finding a minimal (irreducible, or of least order) realization of the form (4) (or (19)).

First note that there are methods to determine the order n of a minimal realization directly from $H(s)$ via the characteristic polynomial and notions such as McMillan degree of $H(s)$ or via the Markov parameters of $H(s)$ and the Hankel matrix. This can be done without finding a minimal realization. Knowing the order of a minimal realization in advance is useful as it provides a guide as to what we should expect when we determine an actual realization. Details may be found in the references below.

In special cases, it is possible that the realization algorithm results directly in a controllable and observable and therefore minimal realization. It is more common however for the algorithm to result in just an either controllable or observable realization, in which case an extra step is needed to isolate the uncontrollable, say, part of the realization and take only the part that it is both controllable and observable. The reader should consult any of the references below for detailed descriptions of several realization algorithms.

Here an example is given of a single-input, single-output system where the resulting realization is controllable and observable, therefore minimal.

Example 1

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}.$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

$$C = \begin{bmatrix} b_0, & b_1, & b_2 \end{bmatrix},$$

is controllable ((A, B) have a form called *controller form*) and observable and therefore minimal realization of $H(s)$; note that all cancellations are assumed to have already taken place between numerator and denominator of a transfer function $H(s)$. This algorithm easily generalizes to the case when the degree of the denominator of $H(s)$ is n (in this example it is 3). Note that if $\lim_{s \to \infty} H(s) = D \neq 0$, then apply the previous algorithm to $\hat{H}(s) = H(s) - D$ to obtain A, B, and C.

Summary

The state variable realization of impulse responses and transfer functions was one of the early problems addressed by system theory. Its solution provides clear understanding of the relations between external (input–output) and internal descriptions of systems. A key result is that any minimal order realization is controllable and observable. Many realization algorithms may be found in the literature. Extensions to polynomial matrix descriptions can also be found in the literature, as well as extensions to partial realizations.

Cross-References

- ▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions
- ▶ Linear Systems: Continuous-Time, Time-Varying State Variable Descriptions
- ▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions
- ▶ Linear Systems: Discrete-Time, Time-Varying, State Variable Descriptions
- ▶ Linear Systems: Continuous-Time Impulse Response Descriptions
- ▶ Linear Systems: Discrete-Time Impulse Response Descriptions

Recommended Reading

A clear understanding of the relationship between external and internal descriptions of systems is one of the principal contributions of systems theory. This topic was developed in the early 1960s with original contributions from Gilbert (1963) and Kalman (1963). The role of controllability and observability in minimal realization is due to Kalman (1963); see also Kalman et al. (1969). For extensive historical comments, see Kailath (1980). The time-varying case is treated in Brockett (1970), Antsaklis and Michel (2006), and Rugh (1996).

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Brockett RW (1970) Finite dimensional linear systems. Wiley, New York

Gilbert E (1963) Controllability and observability in multivariable control systems. SIAM J Control 1:128–151

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Kalman RE (1963) Mathematical description of linear systems. SIAM J Control 1:152–192

Kalman RE, Falb PL, Arbib MA (1969) Topics in mathematical system theory. McGraw-Hill, New York

Moore BC (1981) Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Trans Autom Control AC-26:17–32

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Real-Time Optimization of Industrial Processes

Jorge Otávio Trierweiler

Group of Intensification, Modelling, Simulation, Control and Optimization of Processes (GIMSCOP), Department of Chemical Engineering, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil

Synonyms

RTO

Abstract

RTO aims to optimize the operation of the process taking into account economic terms directly. There are several fundamental gears for smooth operating of an RTO solution. The RTO loop is an extension of feedback control system and consists of subsystems for (a) steady-state detection, (b) data reconciliation and measurement validation, (c) process model updating, and (d) model-based optimization followed by solution validation and implementation. There are several alternatives for each one of these subsystems. This contribution introduces some of the currently used approaches and gives some perspectives for future works in this area.

Keywords

Introduction

Effectiveness, efficiency, product quality, process safety, and low environmental impact are the main driving forces for the improvement of the operation of industrial processes. Real-time (or online) optimization (RTO) is one of the options that are available to achieve these goals and is attracting considerable industrial interest due to its direct and indirect benefits.

RTO systems are model-based, closed-loop process control systems whose objective is to maintain the process operation as nearly as possible to the optimal operating point. Such RTO systems use rigorous process models and current economic information to predict the optimal process operating conditions. Additionally, RTO can mitigate and reject long-term disturbances and performance losses (e.g., due to fouling of heat exchangers or deactivation of catalysts).

The direct benefit from applying RTO is improving the economic performance in terms of increasing the profit of the plant and reducing energy consumption and pollutant emissions.

These are also called the *online benefits*. The indirect benefits result from the tools used in the implementation of RTO. For instance, a better understanding of the processes can be employed to debottleneck the plant and to reduce operating difficulties. In addition, abnormal measurement information obtained from gross error detection can help instrumentation and process engineers to troubleshoot the plant instrument errors. Parameter estimation is very useful for process engineers to evaluate the equipment conditions and to identify decreasing efficiencies and other sources of problems. Furthermore, the detailed process simulation of the model used in online optimization can be used for process monitoring and serve as a training tool for new operators. Finally, the rigorous process model can be used for process maintenance, advanced process control, process design, facility planning, and process monitoring.

Real-time optimization (RTO) solutions have been developed since the early 1980s, and nowadays there are many petrochemical and chemical applications, especially in the production of ethylene and propylene in fluid catalytic cracking units (FCCUs) (Darby et al. 2011). Other successful industrial applications are mentioned in Alkaya et al. (2009) with the respective economic returns.

Control Layers and the RTO Concept

Usually the process control is stratified into several layers, which have different response times and control objectives. RTO is located in an intermediate layer that provides the connection between plant scheduling (medium-term planning) and the control system (short-term process performance). In a plant control hierarchy, process disturbances are controlled using process controllers, whereas the RTO system must track changes in the optimum operating conditions caused by low-frequency process changes (e.g., raw material quality and composition, catalyst deactivation).

The typical structure of an RTO system is shown in Fig. 1, which depicts the elements of the closed-loop system. The RTO loop is

R
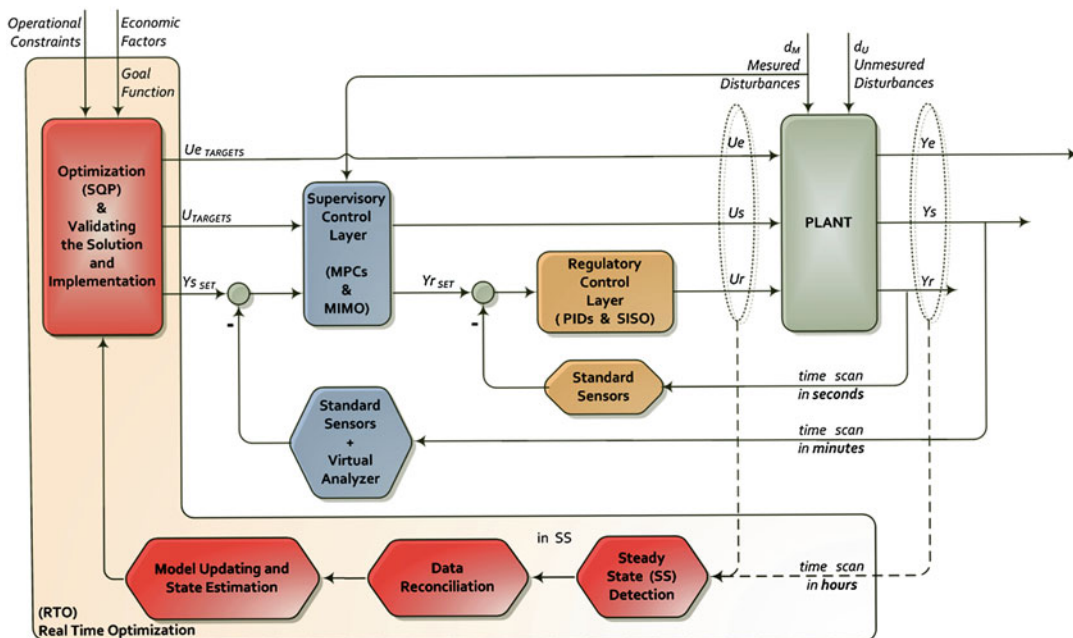
Real-Time Optimization of Industrial Processes, Fig. 1 Basic structure of the traditional RTO

an extension of the feedback control system and consists of subsystems for (a) steady-state detection, (b) data reconciliation and measurement validation, (c) process model updating, and (d) model-based optimization followed by solution validation and implementation. Once the plant operation has reached a steady state, the plant data ($y = [Ye, Ys, Yr]$) are gathered and validated to detect and correct gross errors in the process measurements, and at the same time the measurements may be reconciled using material and energy balances to ensure that the data set used for model updating is consistent. These validated measurements are used to estimate model parameters (θ) to ensure that the model represents the plant faithfully at the current operating point. Then, the optimum controller set points (Ys_{SET}) and manipulated targets ($U_{TARGETS}$ and $Ue_{TARGETS}$) are calculated using the updated model and are transferred to the advanced process controllers after they have been validated to be effectively applied.

Each layer in Fig. 1 has its own specific tasks as discussed in the following:

1. **Regulatory layer.** This layer is focused on basic (e.g., temperature, flow rate) and inventory (e.g., level and pressure) control ensuring safety and operational stability for the industrial plant. The holdups of vapors and gases are measured by pressure sensors, while the holdups of liquids and solids are measured by level and weighing sensors. In the case of unstable processes, the regulatory layer is also responsible for their stabilization, e.g., by temperature control of industrial reactors. No industrial process can operate without this control layer. The typical operation time scan is in the order of seconds. For its design, typical questions that have to be answered are the following: "How to ensure safe unit operation?" "How to quickly meet the demands coming from the supervisory layer or from the operators?" "How to prevent disturbances to propagate throughout the plant?" The control technology that prevails in this layer is SISO (single-input-single-output) PID controllers, with very few cases where the derivative action is effectively employed. In Fig. 1, Yr are the controlled variables of this

layer (e.g., levels, flow rates, pressures, temperatures, pH), and *Ur* are the corresponding manipulated variables, typically control valves.

2. **Supervisory layer.** This layer is concerned with the quality of the final product. The goal is to ensure the specifications without infringing the operating limits of the equipment. Typically, in this layer there is a strong interaction between the controlled variables, requiring tailored multi-look control structures or the use of multivariate control techniques. The dominant advanced technology in this layer is model predictive control (MPC). In this layer the calculations and updates are performed on the time scale of minutes and the typical associated question is "how to ensure the quality of the final product while satisfying the operating constraints and improving the profitability by reducing the variability of the product parameters?" Here the controlled variables (*Ys*) are usually related to the product quality and the manipulated variables are the set points for the regulatory layer Yr_{SET} and additional manipulated variables (Us) not used by the regulatory layer (e.g., variable frequency drive).

3. **RTO layer.** Here the main focus is the profitability of the process. Specifications and operating points (i.e., set points and targets for the manipulated variables) are determined by solving an optimization problem that aims at maximizing the profitability of the process under stationary conditions. When the optimal operating point is close to the operational limits, the real-time optimization is quite straightforward, since it is enough to take the process to these limits, which is usually done by solving a linear programming (LP) optimization problem. Such simple solutions are effective especially in cases where it is known that to maximize or to minimize the flow rate of a given stream will maximize the profitability. As this kind of solution can be easily implemented, most commercial predictive controllers already have an LP or QP layer integrated, using as a model the gain of the dynamic model used in the MPC. However, for processes with large recycling streams and pronounced nonlinearity, this type of solution is not enough to bring the system to its optimal operating conditions. In this case, it is essential to use a nonlinear optimizer that aims at driving the system to operate in the best operating region. When the industrial process works essentially in steady state, the problem can be solved using stationary models. The solutions offered on the market typically involve the use of a stationary process simulator (e.g., Aspen Plus, PRO II). The RTO sampling times are in the time scale of hours and the questions to be answered are the following: "What is the best way of operation?" "How to increase the profitability of the process?" "How to decrease energy consumption and to increase the process efficiency?"

Four Elements of Classical Real-Time Optimization

A standard RTO solution requires that all four calculation blocks illustrated in Fig. 1 work together smoothly. In fact, each block can be formulated as an optimization problem by itself. Sometimes these optimization problems are combined together. Below the alternative techniques that can be applied to each of these subsystems are discussed.

Steady-State Detection (SSD)

As indicated in Fig. 1, the RTO loop execution begins with the detection of a steady state. Identifying a steady state may be difficult because process variables are noisy and measurements do not settle at a constant value. Being at a steady state can be defined as an acceptable constancy of the measurements over a given period of time. Therefore, tests for stationarity are commonly based on checking the constancy of the measured quantities.

Mejía et al. (2010) compared 6 different approaches to SSD using 5,760 simulated data sets. They concluded that the method based on the estimation of the absolute value of the first and the second derivatives defined by

$$I_{DER} = \left| \widehat{\frac{dy}{dx}} \right| + 10 \left| \widehat{\frac{d^2y}{dt^2}} \right| \qquad (1)$$

gives the best results. Although this idea is quite simple, as being at a steady state means zero derivatives by definition, it has some implementation issues, due to signal noise and outliers. These problems can be reduced by smoothing the plant data using smoothing splines, noncausal Butterworth filters, or wavelet decompositions. The second best compared approach was the local autocorrelation (Mejía et al. 2010) followed by the two statistical nonparametric tests of independence hypothesis proposed by Bebar (2005) and by the method of Cao and Rhinehart (1995).

Data Reconciliation (DR)

Within the mathematical models of industrial processes, the balance equations that result from conservation laws of mass, energy, etc., are the core that cannot be subject to debate. If the measured data do not satisfy the balance equations, this fact must be attributed to measurement errors or to fundamental model inadequacies. Ruling out the latter, as measurement errors are always present, before using the measured data, they should be adjusted to obey the conservation laws and other constraints, e.g., of their ranges. The adjustment using optimization techniques combined with the statistical theory of errors is called data reconciliation. Unfortunately the adjustment of all variables can be greatly affected by "gross errors" in one variable, so such errors must be detected.

The relationship between a measurement of a variable and its true value can be represented by

$$y_m = y + \overbrace{e_r + e_g}^{error} \qquad (2)$$

where y_m and y are the measured and true values, while e_r and e_g are the random and gross errors, respectively. The random errors (e_r) are assumed to be zero mean and normally distributed (Gaussian), since they are the result of the simultaneous effect of several causes. The gross errors (e_g) are caused by large nonrandom events. They can

be subdivided into measurement-related errors, such as malfunctioning sensors (e.g., incorrect calibration, sensor degradation, or damage to the electronics), and processes-related errors, such as process leaks.

In the absence of gross errors, the simplest version of data reconciliation can be stated as a quadratic programming (QP) problem

$$\min_y \frac{1}{2}(y - y_m)^T Q^{-1}(y - y_m) \qquad (3)$$

subject to the linear or linearized constraint related to the process model, written as

$$A \cdot y - c = 0.$$

The covariance matrix (Q), which is usually diagonal, captures the variance of the sensors and is responsible to distribute the errors among the measurements (y_m). The solution of this problem is the reconciled value that for this simple case is given analytically by

$$y = \left[I - QA^T (AQA)^{-1} A \right] y_m + QA^T (AQA^T)^{-1} c.$$

A rigorous formulation of the reconciliation problem is possible even with nonlinear constraints; only the general existence and uniqueness of a solution is not warranted theoretically.

Several statistical tests have been constructed for the detection of gross errors. Some of them are based in the distribution of the constraint residuals, i.e., $r_c = A \cdot y_m - c$, and others are based on the distribution of the estimated error after the reconciliation procedure, i.e., $\hat{e} = y_m - y$. The evaluation of r_c does not require solving previously the associated data reconciliation problem. For a complete discussion and review, see Narasimhan and Jordache (1999) and Sequeira et al. (2002).

Model Updating

A key, yet difficult, decision in model parameter adaptation is to select the parameters that are

adapted. These parameters should be identifiable and represent actual changes in the process, and their adaptation should help to approach the true process optimum. Clearly, the smaller the subset of parameters, the better the confidence in the parameter estimates and the lower the required excitation (also the better the regularization effect). But too few adjustable parameters can lead to misleading models and thus wrong proposals for operational changes.

In general, the parameter estimation and updating are limited not only by the lack of information available from experimental data but also by the correlation between the parameters that are identified. The estimation of correlated parameters leads to a high degree of uncertainty in the model, since different combinations of parameter values lead to the same value of the objective function in the estimation problem.

The selection of the right number of parameters to be identified can be done by the analysis of the sensitivity matrix (S). The elements of S, s_{ij}, are the partial derivatives of the measurement y_i with respect to the parameter θj evaluated at the current value of the parameter (θ_0), i.e.,

$$s_{ij} = \left(\frac{\partial y_i}{\partial \theta_j} \right) \Bigg|_{\theta = \theta_0}. \qquad (4)$$

In general, different parameters and measurements have distinct magnitudes. Therefore, scaling is a key issue that has a strong impact on the results. Traditionally each element of the matrix S is scaled by the initial guess θ_{0j} of the parameter and by the average value of the measurement i ($\overline{y_{si}}$). The scaled elements $s_{s,ij}$ are then given by

$$s_{s,ij} = \left(\frac{\theta_{0j}}{\overline{y_{si}}} \right) \left(\frac{\partial y_i}{\partial \theta_j} \right) \Bigg|_{\theta = \theta_0} \qquad (5)$$

This scaling procedure has some problems, once it requires both a good initial guess for the parameters and representative average values for the measurements. But the main drawback is that it does not consider the multivariable nature existing among all parameters and outputs. To solve these drawbacks, Botelho et al. (2012) proposed

to apply diagonal scaling matrices L and R that result from the solution of the convex optimization problem to find the minimized condition number of the sensitivity matrix, $\gamma(LSR)$, i.e.,

$$\begin{aligned} &\min_{L,R} \gamma(LSR) \\ &s.t. L \in \Re^{ny X ny}, \textit{diagonal and nonsingular} \quad (6) \\ &\quad R \in \Re^{n\theta X n\theta}, \textit{diagonal and nonsingular} \end{aligned}$$

This convex optimization problem can be solved using the LMI (linear matrix inequality) approach as described by Boyd et al. (1994). With the optimized scaling matrices L and R, the scaled sensitivity matrix S_s is given by

$$S_s = LSR \qquad (7)$$

With S_s, the best subset of parameters to be estimated can be determined using the non-square relative gain array matrix (NSRGA) as also proposed by Botelho et al. (2012). The NSRGA can be easily calculated for the scaled sensitivity matrix by

$$NSRGA\,(S_s) \underset{=}{\mathrm{def}} S_s \circ (S_s^\dagger)^T \qquad (8)$$

where $(S_s^\dagger)^T$ is the transpose of the pseudo-inverse of S_s and $\circ$ is the entrywise product (also known as the Hadamard or element-wise product). The rows of $NSRGA\,(S_s)$ are related to the output measurements, whereas the columns are related to the parameters. The sum of the values in each column, whose values can vary between 0 and 1, reflects the relevance of each parameter, and it can be used to sort in descending order their influence on the outputs. When the sum of a column is close to 1, the corresponding parameter has a small correlation with the other ones and a strong influence on the output measurements.

Figure 2 illustrates the typical ordering produced by sorting the $NSRGA\,(S_s)$ in descending order. Thus, it is possible to have an idea of which parameters should be selected for estimation. The values presented in this figure suggest that the parameters $P11$ and $P4$ have very small correlation with the others and should be selected as updated parameters, whereas the parameters $P12$
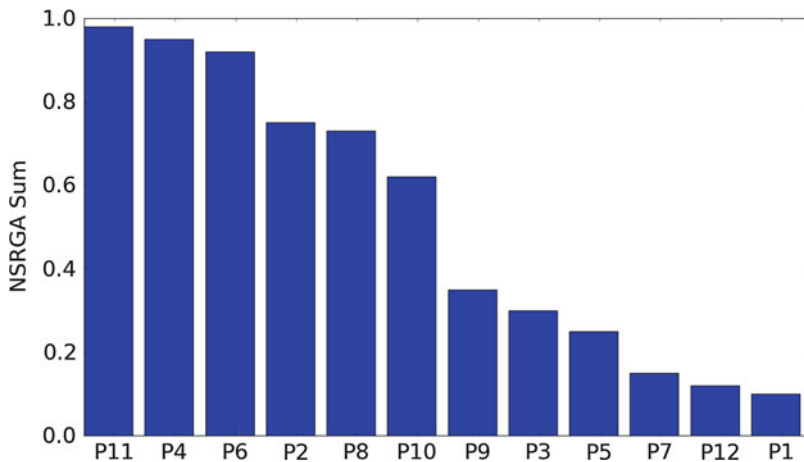
Real-Time Optimization of Industrial Processes, Fig. 2 Illustrative example of using $NSRGA\,(S_s)$ to rank the parameters

and $P1$ show the opposite behavior and should not be updated.

Solving the Optimization Problem

Nonlinear programs for RTO can be formulated using models of different complexities. For example, RTO can be based on process models similar to those used for design and analysis, using commercial simulators (e.g., Aspen Plus, PRO II, HYSYS, etc.). On the other hand, because these problems need to be solved at regular intervals (at least every few hours), detailed simulation models can be partly replaced by correlations or operating curves that are fitted to the process and updated on a longer time scale.

If a rigorous process model is used, the number of nonlinear equations can be very large. The model is usually built by linking smaller submodels. The optimization problem can be formulated as the following nonlinear programming problem (NLP):

$$\min_{x_M, S_M} f(x_M, S_M)$$
$$s.t.$$
$$M_1(x_{M1}, S_{M1}; \theta_{M1}) = 0$$
$$\vdots$$
$$M_n(x_{Mn}, S_{Mn}; \theta_{Mn}) = 0$$
$$OC(x_M, S_M) \leq 0$$
$$x_M = [x_{M1}, \ldots, x_{Mn}], S_M = [S_{M1}, \ldots, S_{Mn}],$$
$$(9)$$

where M_i are the unit modules that can be solved by a tailored procedure in the modular approach or all together in the equation-oriented approach. Each unit model M_i has internal variables x_{Mi} and parameters θ_{Mi}. These unit models are connected by the input and output streams S_{Mi}. Additionally, there are operating constraints OC to capture the possible lower and upper bounds and other equipment constraints. The objective function $f(x_M, S_M)$ is based on an economic model that involves the raw materials, products, and operating costs.

Successive quadratic programming (SQP) has become the most popular method for solving these nonlinear constrained optimization problems. SQP converges the equality and inequality constraints simultaneously with the optimality conditions. This strategy requires relatively few function evaluations and often performs efficiently for process optimization problems. The NLP solver can be implemented in a nonintrusive way, similar to recycle convergence modules that are already in place. As a result, the structure of the simulation environment and the unit operation blocks does not need to be modified in order to include the optimization, so that SQP can be easily incorporated within existing modular simulators and therefore be applied directly to flow sheets modeled in these commercial simulators. However, in this case derivative

information must be obtained by numerical differentiation which increases the effort and slows down convergence near the optimum.

For fully equation-oriented models with the exact first and second derivatives for all constraints and objective functions, efficient NLP algorithms were developed. For instance, large equation-based models can be solved efficiently with structured barrier NLP solvers (see Biegler 2010 for a detailed overview). But for problems where function evaluations are expensive, and gradients and Hessians are difficult to obtain, it is not clear that large-scale NLP solvers should be applied. Black-box optimization models with inexact (or approximated) derivatives and few decision variables are poorly served by large-scale NLP solvers, and derivative-free optimization algorithms should be considered instead. For the standard RTO problems formulated using modular process simulator model, SQP and reduced-space SQP methods are expected to perform well (see Alkaya et al. 2009; Biegler 2010 for detailed discussion).

After solving the RTO optimization problem, it is necessary to decide if the solution can be implemented. For this, it is necessary to verify if the dominant cause of the plant changes is noise, since in this case implementing these changes could lower the profit. Thus, an important challenge in RTO results analysis is to determine when to implement the calculated changes (Miletic and Marlin 1996).

Summary and Future Directions

RTO aims at optimizing the operation of the process taking into account economic terms directly. There are several fundamental needs for a smooth operation of an RTO solution. The central point is the mathematical modeling which can be a complex first principle model or be based on simple operating curves. If a good model is available, it is necessary to have a good characterization of the inlet streams (properties and composition), to employ data reconciliation and gross error detection and steady-state identification. Finally, the efficiency of the optimizer is a key issue.

Due to the time and resources needed to implement and maintain an RTO solution, a full RTO project involves a certain high risk. Therefore, in cases where simpler and easier approaches can be applied with equivalent economic benefits, they should be used instead. For processes with large recycle streams, it is worthwhile to apply the classical RTO strategy, i.e., the one discussed in the last section. In this case the optimal solution is not trivial, once it is not simply the maximal capacity of the plant. For the cases where the optimal operating constraint is a direct consequence of the operating process capacity, the economic optimization can be easily included in the LP or QP layer implemented usually within a model predictive controller.

In the previous section, the so-called two-step approach, where the measurements are used to refine the process model, which is then used to repeat the optimization, was described. Several RTO schemes have emerged since the development of this two-step approach in the 1970s. Recently, it has been proposed to update the model differently. Instead of adjusting the model parameters, one updates correction terms that are added to the cost and constraint functions of the optimization problem. The technique, labeled as modifier adaptation (RTO-MA), forces the modeled cost and constraints to match the plant values (Gao and Engell 2005; Marchetti et al. 2009). The main advantage of RTO-MA compared to the two-step approach lies in its ability to converge to the true plant optimum, even in the presence of structural plant-model mismatch. RTO-MA is a static optimization method which means that its application to a continuous process requires waiting for reaching the steady state before taking measurements, updating the correction terms, and repeating the numerical optimization. Hence, several iterations are generally required before convergence can be achieved. In contrast, implicit methods, such as self-optimizing control (Skogestad 2000) and NCO tracking (François et al. 2005), propose to adjust the inputs online in a control-inspired manner. Especially simple to be implemented is the "self-optimizing" approach, where a feedback control structure is chosen so that maintaining

some function of the measured variables constant automatically maintains the process near an economically optimal steady state in the presence of disturbances. The problem is posed from a plantwide perspective, since the economics are determined by overall plant behavior.

The classical steady-state RTO has some drawbacks related to its low frequency of execution. It is normally run twice or three times per day and one does not consider the cost of transiting from one operating condition to another. Some plants need to respond to market changes very quickly, like grade changes in polymerization and petroleum process. In these processes, market competition requires the capability to accommodate fast and cost-effective transitions so that companies can produce and sell on demand at favorable prices. To provide this capability, dynamic RTO is being developed and implemented in industrial processes. The largest difference between steady-state and dynamic RTOs is that traditional RTO only provides optimal operating conditions at the steady state, while dynamic RTO provides a trajectory of changes of operating conditions. Dynamic RTO does not require steady-state conditions to be applied. The formulation and solution of the problem DRTO are very similar to the approach used to solve nonlinear predictive controllers (NMPC), with the primary difference the inclusion of economic aspects in the objective function (Engell 2007).

Cross-References

▶ Control Structure Selection
▶ Industrial MPC of Continuous Processes
▶ Model-Based Performance Optimizing Control
▶ Model-Predictive Control in Practice

Recommended Reading

As a number of design decisions must be made in the construction of a RTO system, there is no single approach how to implement it. The elements of the solution were discussed here which should be viewed as a starting point for further reading. The review paper by Engell (2007) discusses and compares several approaches for RTO and DRTO giving a quite general and broad perspective of the area. For the reader interested more in the solution and formulation of the optimization problems, the book by Biegler (2010) is a very good starting point and gives a complete discussion about the solvers currently used, illustrating the application with several examples. For data reconciliation and gross error detection the book by Narasimhan and Jordache (1999) is a good starting point. Finally, an industrial discussion about RTO and alternative approaches that have been used in the industry can be found in Darby et al. (2011).

Bibliography

Alkaya D, Vasamtharajan S, Biegler LT (2009) Successive quadratic programming: applications in the process industry. In: Floudas CA, Pardalos PM (eds) Encyclopedia of optimization. Springer, Berlin, pp 3853–3864

Bebar M (2005) Regelgütebewertung in kontinuierlichen verfahrenstechnischen Anlagen. Ruhr-Universität Bochum

Biegler L (2010) Nonlinear programming: concepts, algorithms, and applications to chemical processes. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics: Mathematical Optimization Society, Philadelphia

Botelho VR, Trierweiler LF, Trierweiler JO (2012) A new approach for practical identifiability analysis applied to dynamic phenomenological models. Paper presented at the International symposium on advanced control of chemical processes – ADCHEM 2012, Singapura

Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. Studies in applied and numerical mathematics. Society for Industrial and Applied Mathematics, Philadelphia

Cao S, Rhinehart RR (1995) An efficient method of on-line identification of steady state. Rev J Process Control 5:11

Darby ML, Nikolaou M, Jones J, Nicholson D (2011) RTO: an overview and assessment of current practice. Rev J Process Control 21(6):874–884. doi:http://dx.doi.org/10.1016/j.jprocont.2011.03.009

Engell S (2007) Feedback control for optimal process operation. Rev J Process Control 17(3):203–219. doi:http://dx.doi.org/10.1016/j.jprocont.2006.10.011

François G, Srinivasan B, Bonvin D (2005) Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. Rev J Process Control 15(6):701–712. doi:http://dx.doi.org/10.1016/j.jprocont.2004.11.006

Gao W, Engell S (2005) Iterative set-point optimization of batch chromatography. Rev Comput Chem Eng 29(6):1401–1409. doi:http://dx.doi.org/10.1016/j.compchemeng.2005.02.035

Marchetti A, Chachuat B, Bonvin D (2009) Modifier-adaptation methodology for real-time optimization. Rev Ind Eng Chem Res 48(13):6022–6033. doi:10.1021/ie801352x

Mejía RIG, Duarte MB, Trierweiler JO (2010) Avaliação do desempenho e ajuste automático de métodos de identificação de estado estacionário. In: ABEQ (ed) COBEQ 2010 Congresso Brasileiro de Engenharia Química, 10, Foz do Iguaçú

Miletic I, Marlin T (1996) Results analysis for real-time optimization (RTO): deciding when to change the plant operation. Rev Comput Chem Eng 20(Supplement 2):S1077-S1082. doi:http://dx.doi.org/10.1016/0098-1354(96)00187-1

Narasimhan S, Jordache C (1999) The importance of data reconciliation and gross error detection-1. In: Data reconciliation and gross error detection. Gulf Professional, Burlington, pp 1–31

Sequeira SE, Graells M, Puigjaner L (2002) Real-time evolution for on-line optimization of continuous processes. Rev Ind Eng Chem Res 41(7):1815–1825. doi:10.1021/ie010464l

Skogestad S (2000) Plantwide control: the search for the self-optimizing control structure. Rev J Process Control 10(5):487–507. doi:http://dx.doi.org/10.1016/S0959-1524(00)00023-8

Redundant Robots

Stefano Chiaverini
Dipartimento di Ingegneria Elettrica e dell'Informazione "Maurizio Scarano", Università degli Studi di Cassino e del Lazio Meridionale, Cassino (FR), Italy

Abstract

Redundancy may occur in different ways in a robotic system. This entry focuses on the resolution of kinematic redundancy, i.e., on the techniques for exploiting the redundant degrees of freedom in the solution of the inverse kinematics problem; this is indeed an issue of major relevance for motion planning and control purposes.

Keywords

Introduction

Redundant robots possess more resources than those strictly required to execute their task; this provides the robot with an increased capacity of facing real-world applications by allowing to handle performance issues besides the mere achievement of a given motion trajectory.

Redundancy may occur in the sensory system, in the mechanical structure, and/or in the actuation system, thus allowing, e.g., fault accommodation, multisensory perception, dexterous motion, and load sharing. Nevertheless, unless otherwise specified, by *redundant robot* it is meant one that has a kinematically redundant mechanical structure, i.e., provided with more degrees of freedom than those strictly required to execute its task; this also typically leads to a redundancy in the number of actuators and sensors. Noticeably, *kinematic redundancy* is usually the key to handle the avoidance of singular configurations, the occurrence of joint limits, the engagement of obstacles in the workspace, and the minimization of joint torques or energy. In practice, if properly managed, the increased dexterity characterizing kinematically redundant robots may allow them to achieve a higher degree of autonomy.

In principle, no robot is inherently redundant; rather, there are certain tasks with respect to which it may become redundant. Nevertheless, since most papers in the classical literature on the topic have dealt with robotic manipulators (for which a general task consists in tracking an end-effector motion trajectory requiring six degrees of freedom), a robot arm with seven or more joints is often considered as a typical example of an inherently redundant manipulator. However,

even robot arms with fewer degrees of freedom, like conventional six-joint industrial manipulators, may become kinematically redundant for specific tasks, such as simple end-effector positioning without constraints on the orientation.

In the case of traditional industrial applications involving nonredundant mechanical structures, the occurrence of singular configuration and/or the presence of obstacles in the workcell resulted in the need of a carefully structured (and static) working space where the motion of the manipulator could be planned in advance.

On the other hand, the presence of redundant degrees of freedom allows motions of the manipulator that do not displace the end effector (the so-called self-motions or internal motions); this implies that the same end-effector task can be executed with several different joint motions, giving the possibility of better exploiting the workspace of the manipulator and ultimately resulting in a more versatile robotic arm (see Fig. 1). Such feature is a key to allow operation in unstructured and/or dynamically varying environments that characterize advanced industrial applications and service robotics scenarios.

The biological archetype of a robotic manipulator is the human arm, which, not surprisingly, also inspires the terminology used to characterize the serial-chain structure of a robot *arm*. Remarkably, a simple look at the human arm kinematics from the torso to the hand allows to recognize seven degrees of freedom (three at the *shoulder*,

one at the *elbow*, and three at the *wrist*) that make a manipulator kinematically redundant.

The kinematic arrangement of the human arm has been replicated in a number of robots often termed as *human-armlike manipulators* (see, e.g., Fig. 2). Manipulators with a larger number of joints are often called *hyperredundant robots* and include – among others – snakelike robots (Fig. 3).

The use of two or more robotic structures to execute a task (as in the case of cooperating manipulators or multifingered hands or multiarm/multilegged robots) also gives rise to kinematic redundancy. A headed multilimb structure is typical of a humanoid robot (Fig. 4). Redundant mechanisms also include vehicle-manipulator systems (Fig. 5).

Although the realization of a kinematically redundant structure raises a number of issues from the point of view of mechanical design, this entry focuses on the techniques for exploiting the redundant degrees of freedom in the solution
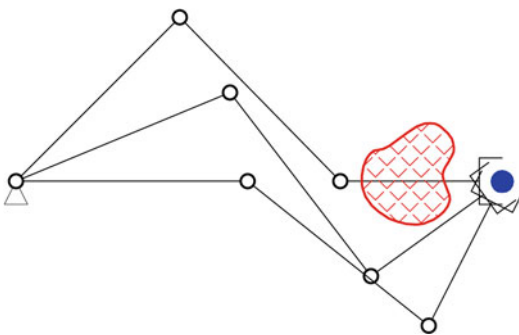
Redundant Robots, Fig. 1 A self-motion of the arm that keeps the end-effector positioned at the *blue* spot. It is possible to choose configurations that both take the *blue* spot and avoid the *red* obstacle
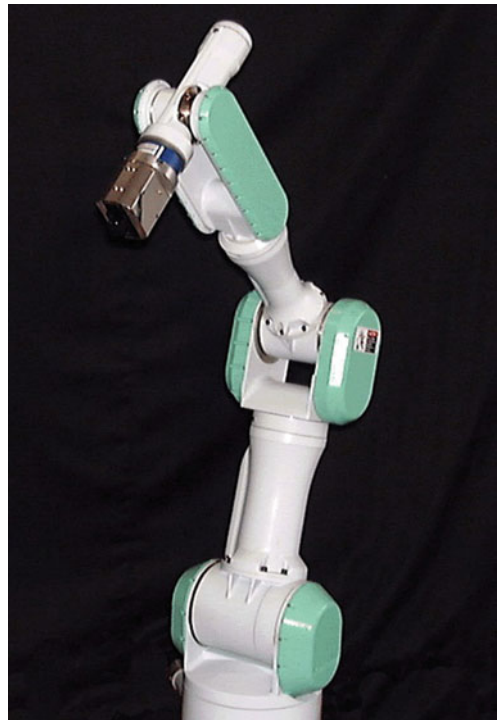
Redundant Robots, Fig. 2 The Mitsubishi PA-10 manipulator

Redundant Robots, Fig. 3 The SnakeRobots.com S7 snake robot prototype

Redundant Robots, Fig. 5 The KUKA youBot

Redundant Robots, Fig. 4 The Honda ASIMO

of the inverse kinematics problem. This is an issue of major relevance for motion planning and control purposes.

Task-Oriented Kinematics

The relationship between the N variables representing the configuration q of an articulated manipulator in the *joint space* and the M variables describing an assigned task t in an appropriate *task space* constitutes a task-oriented kinematics; this can be established at the position, velocity, or acceleration level. Typically, one has $N \geq M$, so that the joints can provide at least the degrees of freedom required for the end-effector task. If $N > M$ strictly, the manipulator is *kinematically redundant*.

At the position level, the *direct kinematics* equation takes on the form

$$t = k_t(q), \tag{1}$$

where k_t is a nonlinear vector function.

Besides the direct kinematics expressed at the position level, it is useful to consider the *first-order differential kinematics* (Whitney 1969)

$$\dot{t} = J_t(q)\,\dot{q}, \tag{2}$$

that can be obtained by differentiating Eq. (1) w.r.t. time. In (2), the mapping between the task-space and the joint-space velocities is held by the $M \times N$ *task Jacobian* matrix $J_t(q) = \partial k_t / \partial q$ (also called *analytic Jacobian*).

Remarkably, $\dot{t}$ expresses the rate of change of the variables adopted to describe the task and thus does not necessarily have the meaning of an end-effector velocity. In general, by denoting the end-effector spatial velocity v_N as the stack of the 3D

translational and angular end-effector velocities, the following relationship holds

$$\dot{\boldsymbol{t}} = \boldsymbol{T}(t)\,\boldsymbol{v}_N\,, \qquad (3)$$

where $\boldsymbol{T}$ is an $M \times 6$ transformation matrix.

For a given manipulator, the mapping

$$\boldsymbol{v}_N = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \qquad (4)$$

relates a joint-space velocity to the corresponding end-effector velocity through the $6 \times N$ *geometric Jacobian* matrix $\boldsymbol{J}$.

By comparing (2)–(4), the relation between the geometric Jacobian and the task Jacobian can be found as

$$\boldsymbol{J}_\text{t}(\boldsymbol{q}) = \boldsymbol{T}(t)\,\boldsymbol{J}(\boldsymbol{q})\,. \qquad (5)$$

Further differentiation of (2) w.r.t. time provides the following relationship between the acceleration variables:

$$\ddot{\boldsymbol{t}} = \boldsymbol{J}_\text{t}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_\text{t}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\,. \qquad (6)$$

This equation is also known as *second-order differential kinematics*.

Singularities

A robot configuration $\boldsymbol{q}$ is *singular* if the task Jacobian matrix is rank deficient at it. Considering the role of $\boldsymbol{J}_\text{t}$ in (2) and (6), it is easy to realize that at a singular configuration, it is impossible to generate end-effector task velocities or accelerations in certain directions. Further insight can be gained by looking at (5), which indicates that a singularity may be due to a loss of rank of the transformation matrix $\boldsymbol{T}$ and/or of the geometric Jacobian matrix $\boldsymbol{J}$.

Rank deficiencies of $\boldsymbol{T}$ are only related to the mathematical relationship between $\boldsymbol{v}_N$ and $\boldsymbol{t}$, $\dot{\boldsymbol{t}}$; for this reason, a configuration at which $\boldsymbol{T}$ is singular is referred to as a *representation singularity*. A representation singularity is not directly related to the true motion capabilities of the manipulator structure, which can be instead inferred by the analysis of the geometric Jacobian matrix. Rank deficiencies of $\boldsymbol{J}$ are in fact related to loss of mobility of the manipulator end effector; indeed, end-effector velocities exist in this case that are unfeasible for any velocity commanded at the joints. A configuration at which $\boldsymbol{J}$ is singular is referred to as a *kinematic singularity*.

Since redundancy resolution methods involve the inversion of the task differential kinematics (2) and (6), the handling of singularities through proper treatment of the Jacobian matrix is very important. However, due to space limitations, this topic is out of the scope of this entry and in the following, we will assume that the Jacobian matrices at issue are all full rank.

Null-Space Velocities

With a full-rank task Jacobian, at each configuration an $N - M$ dimensional null space of $\boldsymbol{J}_\text{t}$ exists made of the set of joint-space velocities that yield zero task velocity; these are thus called *null-space velocities* in short.

Remarkably, the components of $\dot{\boldsymbol{q}}$ in the null-space of $\boldsymbol{J}_\text{t}$ produce a change in the configuration of the manipulator without affecting its task velocity. This can be exploited to achieve additional goals – like obstacle or singularity avoidance – in addition to the realization of a desired task motion and constitutes the core of redundancy resolution approaches.

Inverse Differential Kinematics

The inverse kinematics problem can be solved by inverting the direct kinematics equation (1), the first-order differential kinematics (2) or the second-order differential kinematics (6). With a time-varying desired task reference, it is convenient to solve the differential kinematic relationships because these represent linear equations with the task Jacobian as the coefficient matrix.

For a kinematically redundant manipulator, the general solution of (2) or (6) can be expressed by resorting to the pseudoinverse $\boldsymbol{J}_\text{t}^\dagger$ of the task Jacobian matrix (Whitney 1969).

The general solution of (2) can be written as

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_\text{t}^\dagger(\boldsymbol{q})\,\dot{\boldsymbol{t}} + \boldsymbol{N}_{\boldsymbol{J}_\text{t}}(\boldsymbol{q})\,\dot{\boldsymbol{q}}_0\,, \qquad (7)$$

where

$$N_{J_t}(q) = I - J_t^\dagger(q)J_t(q)$$

is an orthogonal projection matrix into the null-space of J_t, and $\dot{q}_0$ is an arbitrary joint-space velocity; the second part of the solution is therefore a null-space velocity. The particular solution obtained by setting $\dot{q}_0 = 0$ in (7) is known as the *pseudoinverse solution*.

As for the second-order kinematics (6), its solution can be expressed in the general form

$$\ddot{q} = J_t^\dagger(q)(\ddot{t} - \dot{J}_t(q,\dot{q})\dot{q}) + N_{J_t}(q)\,\ddot{q}_0, \quad (8)$$

where $\ddot{q}_0$ is an arbitrary joint-space acceleration.

In summary, for a kinematically redundant manipulator, the inverse kinematics problem admits an infinite number of solutions, so that a methodology to select one of them is needed.

Redundancy Resolution via Optimization

An approach to redundancy resolution is based on the optimization of suitable performance criteria.

Performance Criteria

The availability of redundant degrees of freedom can be used to improve the value of performance criteria during the motion. These criteria may depend on the robot joint configuration only or involve also velocities and/or accelerations.

The most frequently considered performance objective for trajectory tracking tasks is singularity avoidance. In fact, singularities lead to decreased mobility, and adding kinematic redundancy allows to reduce the extension of the workspace region where the manipulator is necessarily at a singular configuration (*unavoidable* singularities Baillieul et al. 1984). Possible performance criteria to drive the manipulator motion out of avoidable singularities are configuration-dependent functions that characterize the distance from singular configurations, i.e., the manipulability measure,

the condition number, and the smallest singular value of J_t.

Since kinematic inversion produces very high joint velocities in the vicinity of singular configurations, a conceptually different possibility is to minimize the norm of the joint velocity generated by the redundancy resolution scheme.

Redundancy can be also used to keep a robot away from undesired regions of the joint space or of the task space. For example, it might be desired that a manipulator avoids reaching mechanical joint limits (Liégeois 1977). Another interesting application is obstacle avoidance, which can be enforced by minimizing suitable artificial potential functions defined on the basis of the image of the obstacle region in the configuration space.

Many other performance criteria can be found in the literature.

Local Optimization

Equation (7) provides least-squares solutions to the end-effector task constraint (2), so that it minimizes $\|\dot{t} - J\dot{q}\|$.

The simplest form of local optimization is represented by the pseudoinverse solution that provides the joint velocity with the minimum norm among those which realize the task constraint. Clearly, the joint movement generated by this locally optimal solution does not provide global velocity minimization along the entire manipulator motion; therefore, singularity avoidance is not guaranteed (Baillieul et al. 1984).

In terms of the inverse differential kinematics problem, the least-squares property may quantify the accuracy of the end-effector task realization, while the minimum norm property may be relevant for the feasibility of the joint-space velocities.

Another possibility is to use the general solution (7), choosing $\dot{q}_0$ as

$$\dot{q}_0 = -k_H\,\nabla H(q), \quad (9)$$

where k_H is a scalar stepsize and $\nabla H(q)$ denotes the gradient of a scalar configuration-dependent performance criterion H which is desired to minimize (Liégeois 1977).

As for the second-order solution (8), choosing $\ddot{\boldsymbol{q}}_0 = \boldsymbol{0}$ gives the minimum norm acceleration solution.

Global Optimization

Local optimization algorithms can lead to unsatisfactory performance over long-duration tasks. It is therefore natural to consider the possibility of selecting $\dot{\boldsymbol{q}}_0$ in (7) so as to minimize integral criteria of the form

$$\int_{t_i}^{t_f} H(\boldsymbol{q}) \, dt$$

defined over the whole duration of the task. Unfortunately, the solution of these problems (naturally formulated within the Calculus of Variations framework) may not exist and does not admit a closed form in general. One way to make the problem solvable is to use an integral criterion in quadratic form in the joint velocities or accelerations. However, this is more easily done at the second-order kinematic level (see section "Second-Order Redundancy Resolution").

Redundancy Resolution via Task Augmentation

Another approach to redundancy resolution consists in augmenting the task vector so as to tackle additional objectives expressed as constraints.

Extended Jacobian

The extended Jacobian technique (Baillieul 1985) enforces an appropriate number of functional constraints to be fulfilled along with the original end-effector task.

Given an objective function $g(\boldsymbol{q})$, if $\boldsymbol{J}_t$ has full rank a set of $N - M$ independent constraints can be obtained from the equation

$$\left. \frac{\partial g(\boldsymbol{q})}{\partial \boldsymbol{q}} \right|_{\boldsymbol{q} = \widehat{\boldsymbol{q}}} \boldsymbol{N}_{J_t}(\widehat{\boldsymbol{q}}) = \boldsymbol{0}^{\mathrm{T}},$$

where $\widehat{\boldsymbol{q}}$ is the current joint configuration such that the function $g(\boldsymbol{q})$ is at an extreme; these

$N - M$ independent constraints can be written in vector form as

$$\boldsymbol{h}(\widehat{\boldsymbol{q}}) = \boldsymbol{0}.$$

For a motion that tracks a trajectory $\boldsymbol{t}(t)$ by keeping $g(\boldsymbol{q})$ extremized at each time, it is

$$\begin{bmatrix} \boldsymbol{t}(t) \\ \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_t\big(\boldsymbol{q}(t)\big) \\ \boldsymbol{h}\big(\boldsymbol{q}(t)\big) \end{bmatrix},$$

that, similarly to (1) and (2), leads to define an *extended Jacobian* matrix as

$$\boldsymbol{J}_{\mathrm{ext}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_t(\boldsymbol{q}) \\ \dfrac{\partial \boldsymbol{h}(\boldsymbol{q})}{\partial \boldsymbol{q}} \end{bmatrix}.$$

Therefore, if the initial joint configuration extremizes $g(\boldsymbol{q})$ and provided that $\boldsymbol{J}_{\mathrm{ext}}$ does not become singular, the time integral of the inverse mapping

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\mathrm{ext}}^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{t}} \\ 0 \end{bmatrix} \tag{10}$$

tracks the assigned end-effector trajectory $\boldsymbol{t}(t)$ propagating joint configurations that extremize $g(\boldsymbol{q})$.

The extended Jacobian method has a major advantage over the pseudoinverse solution in that it is *cyclic*, i.e., it generates repetitive joint motion from a repetitive task motion. Moreover, solution (10) can be made equivalent to (7) via suitable choice of the vector $\dot{\boldsymbol{q}}_0$ (Baillieul 1985).

Augmented Jacobian

The task-space augmentation approach is based on the direct definition of a constraint task to be fulfilled along with the end-effector task (Sciavicco and Siciliano 1988).

In detail, let $\boldsymbol{t}_c$ collect P variables that describe the additional tasks to be fulfilled besides the end-effector task $\boldsymbol{t}$. In the general case, it is $P \leq N - M$ although full redundancy exploitation suggests to consider exactly $P = N - M$.

The relation between the joint-space and the *constraint-task* coordinates can be considered as a direct kinematics equation

$$t_c = k_c(q),$$

where k_c is a continuous nonlinear vector function. At this point, an *augmented task* can be defined by stacking the end-effector task with the constraint task as

$$t_a = \begin{bmatrix} t \\ t_c \end{bmatrix} = \begin{bmatrix} k_t(q) \\ k_c(q) \end{bmatrix}.$$

According to this definition, finding a joint configuration q that brings t_a at some desired value means to satisfy both the end effector and the constraint task at the same time.

A solution to this problem can be found at the differential level by inverting the mapping

$$\dot{t}_a = J_a(q)\,\dot{q} \qquad (11)$$

where the matrix

$$J_a(q) = \begin{bmatrix} J_t(q) \\ J_c(q) \end{bmatrix}$$

is termed *augmented Jacobian* and $J_c(q) = \partial k_c / \partial q$ is the $P \times N$ *constraint-task Jacobian* matrix.

A particular choice for the constraint-task vector is $t_c = h(q)$, with h defined as explained in section "Extended Jacobian", that allows the augmented Jacobian method to embed the extended Jacobian one.

Algorithmic Singularities

The specification of additional goals besides tracking the end-effector task raises the possibility that configurations exist at which the augmented kinematics problem is singular while the sole end-effector task kinematics is not; these configurations are termed *algorithmic singularities* (Baillieul 1985). With reference to the velocity mappings (10) and (11), an algorithmically singular configuration is one at which the extended and the augmented Jacobians, respectively, are singular while J_t is full rank.

Remarkably, algorithmic singularities arise from the way in which the constraint task conflicts with the end-effector task and are not

a problem of the specific inverse kinematic technique (Baillieul 1985). This is easily understandable in simple situations such as that of a desired trajectory passing through an obstacle, where either the trajectory is tracked or the obstacle is avoided, so that both tasks cannot be achieved together. If the origin of the conflict between the two tasks has a clear meaning, the algorithmic singularity may be avoided by keenly specifying the constraint task case-by-case; otherwise, analytical tools must be adopted.

Task Priority

Conflicts between the end-effector task and the constraint task are handled in the framework of the task-priority strategy by suitably assigning an order of priority to the given tasks and then satisfying the lower-priority task only in the null-space of the higher-priority task (Maciejewski and Klein 1985; Nakamura et al. 1987). The idea is that, when an exact solution does not exist, the reconstruction error should only affect the lower-priority task.

With reference to solution (7), the task-priority method consist in computing $\dot{q}_0$ so as to suitably achieve the P-dimensional constraint-task velocity $\dot{t}_c$. Remarkably, the projection of $\dot{q}_0$ onto the null-space of J_t ensures lower priority of the constraint task with respect to the end-effector task since it results in a null-space velocity for the end-effector task.

Consistently with the defined order of priority between the two tasks, a reasonable choice is then to guarantee exact tracking of the primary-task velocity while minimizing the constraint-task velocity reconstruction error $\dot{t}_c - J_c\dot{q}$; this gives (Maciejewski and Klein 1985)

$$\dot{q} = J_t^{\dagger}(q)\dot{t} + \left(J_c(q)N_{J_t}(q)\right)^{\dagger}\left(\dot{t}_c - J_c(q)J_t^{\dagger}(q)\dot{t}\right). \qquad (12)$$

It can be recognized that the problem of algorithmic singularities still remains; in fact, the matrix $J_c \cdot N_{J_t}$ may lose rank with full-rank J_t and J_c. However, differently from the task-space augmentation approach, correct primary-task solutions are expected as long as the sole

primary-task Jacobian matrix is full rank. On the other hand, out of the algorithmic singularities, the task-priority strategy gives the same solution as the task-space augmentation approach; this implies that close to an algorithmic singularity, the solution becomes ill-conditioned and large joint velocities may result.

Another approach is to relax minimization of the secondary-task velocity reconstruction constraint and simply pursue tracking of the components of $J_{\mathrm{c}}^{\dagger}\dot{t}_{\mathrm{c}}$ that do not conflict with the primary task (Chiaverini 1997), namely,

$$\dot{q} = J_{\mathrm{t}}^{\dagger}(q)\dot{t} + N_{J_{\mathrm{t}}}(q)J_{\mathrm{c}}^{\dagger}(q)\dot{t}_{\mathrm{c}}. \qquad (13)$$

A nice property of solution (13) is that algorithmic singularities are decoupled from the singularities of J_{c}.

Second-Order Redundancy Resolution

Redundancy resolution at the acceleration level allows the consideration of dynamic performance along the manipulator motion. Moreover, the obtained acceleration profiles (together with the corresponding positions and velocities) can be directly used as reference signals of task-space dynamic controllers.

The simplest scheme operating at the acceleration level is represented by (8) with $\ddot{q} = 0$. Similar to the velocity-level pseudoinverse solution, the joint motion generated by this locally optimal solution does not result in global acceleration minimization. Remarkably, provided that the appropriate boundary conditions are satisfied, this solution leads to the minimization of the integral of $\dot{q}^{\mathrm{T}}\dot{q}$ (Kazerounian and Wang 1988).

More flexibility in the choice of performance criteria is obviously obtained by considering the full second-order solution (8). Let the manipulator dynamic model be expressed as

$$\tau = H(q)\ddot{q} + c(q,\dot{q}) + \tau_g(q),$$

where τ is the vector of actuator torques, H is the manipulator inertia matrix, c is the vector of

centrifugal/Coriolis terms, and τ_g is the gravitational torque vector. Choosing the null-space acceleration in (8) as

$$\ddot{q}_0 = -\big(H(q)N_{J_{\mathrm{t}}}(q)\big)^{\dagger}$$
$$\cdot\Big(H(q)J_{\mathrm{t}}^{\dagger}(q)\big(\ddot{t}-\dot{J}_{\mathrm{t}}(q,\dot{q})\dot{q}\big)+c(q,\dot{q})+\tau_g(q)\Big)$$

leads to the local minimization of the actuator torque norm $\tau^{\mathrm{T}}\tau$ (Hollerbach and Suh 1987).

Another interesting inverse solution, which minimizes the integral of the manipulator kinetic energy, is the following Kazerounian and Wang (1988):

$$\ddot{q} = J_{\mathrm{t},H}^{\dagger}(q)\big(\ddot{t} - \dot{J}_{\mathrm{t}}(q,\dot{q})\dot{q}\big)$$
$$+ \big(I - J_{\mathrm{t},H}^{\dagger}(q)J_{\mathrm{t}}(q)\big)H^{-1}(q)c(q,\dot{q}),$$

where the *inertia-weighted task Jacobian pseudoinverse* can be computed as

$$J_{\mathrm{t},H}^{\dagger}(q) = H^{-1}(q)\,J_{\mathrm{t}}^{\mathrm{T}}(q)$$
$$\Big(J_{\mathrm{t}}(q)\,H^{-1}(q)\,J_{\mathrm{t}}^{\mathrm{T}}(q)\Big)^{-1}.$$

Once again, the correct boundary conditions must be used.

Summary and Future Directions

To discuss kinematic redundancy, the concept of task-oriented kinematics has been first recalled with the basic methods for its inversion at the velocity and acceleration level. Next, different methods to solve kinematic redundancy at the velocity level have been arranged in two main categories, namely, those based on the optimization of suitable performance criteria and those relying on the augmentation of the task space. Finally, redundancy resolution methods at the acceleration level have been considered in order to take into account dynamics issues, e.g., torque or kinetic energy minimization.

Besides the classical linear algebra methods and optimization tools still ever under investigation, new methodological approaches to

Redundant Robots, Fig. 6 The DLR rolling justin

redundancy resolution recently include learning algorithms (Rolf et al. 2010) and soft computing techniques (Liu and Li 2006). Active fields of new applications are in sensorial redundancy for data fusion (Luo and Chang 2012) and in systems (like the one in Fig. 6) with a large number of degrees of freedom, namely, hyperredundant robots (Salvietti et al. 2009), humanoids (Kanoun and Laumond 2010), and multirobot systems (Antonelli et al. 2010).

Cross-References

▶ Cooperative Manipulators
▶ Optimal Control and Mechanics
▶ Robot Motion Control

Recommended Reading

Because of space and scope limitations, in drawing on overview of such a mature and well-developed topic, there are a number of techniques

and details that go neglected in any case; a slightly more extensive treatment of kinematic redundancy, including a touch on singularity robustness, cyclicity, and hyperredundant manipulators with related first-reading bibliography can be found in Chiaverini et al. (2008). Other major issues of interest that could not be covered here are in the use of kinematic redundancy for fault tolerance, for improved grasping, and for motion/force control; see, e.g., Roberts et al. (2008), Prats et al. (2011), and Khatib (1990), respectively.

Bibliography

Antonelli G, Arrichiello F, Chiaverini S (2010) Flocking for multi-robot systems via the null-space-based behavioral control. Swarm Intell 4(1):37–56

Baillieul J (1985) Kinematic programming alternatives for redundant manipulators. In: Proceedings of the 1985 IEEE international conference on robotics and automation, St. Louis, pp 722–728

Baillieul J, Hollerbach J, Brockett R (1984) Programming and control of kinematically redundant manipulators. In: Proceedings of the 23th IEEE conference on decision and control, Las Vegas, pp 768–774

Chiaverini S (1997) Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Trans Robot Autom 13:398–410

Chiaverini S, Oriolo G, Walker I (2008) Kinematically redundant manipulators. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer, Berlin, pp 245–268

Hollerbach J, Suh K (1987) Redundancy resolution of manipulators through torque optimization. IEEE J Robot Autom 3:308–316

Kanoun O, Laumond JP (2010) Optimizing the stepping of a humanoid robot for a sequence of tasks. In: Proceedings of the 10th IEEE-RAS international conference on humanoid robots, Nashville, pp 204–209

Kazerounian K, Wang Z (1988) Global versus local optimization in redundancy resolution of robotic manipulators. Int J Robot Res 7(5):3–12

Khatib O (1990) Motion/force redundancy of manipulators. In: Proceedings of the Japan-USA symposium on flexible automation, Kyoto, pp 337–342

Liégeois A (1977) Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans Syst Man Cybern 7:868–871

Liu Y, Li Y (2006) A new method of executing multiple auxiliary tasks by redundant nonholonomic mobile manipulators. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems, Beijing, pp 1–6

R

Luo R, Chang CC (2012) Multisensor fusion and integration: a review on approaches and its applications in mechatronics. IEEE Trans Ind Inform 8:49–60

Maciejewski A, Klein C (1985) Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int J Robot Res 4(3):109–117

Nakamura Y, Hanafusa H, Yoshikawa T (1987) Task-priority based redundancy control of robot manipulators. Int J Robot Res 6(2):3–15

Prats M, Sanz P, del Pobil A (2011) The advantages of exploiting grasp redundancy in robotic manipulation. In: Proceedings of the 5th international conference on automation, robotics and applications, Wellington, pp 334–339

Roberts R, Hyun G, Maciejewski A (2008) Fundamental limitations on designing optimally fault-tolerant redundant manipulators. IEEE Trans Robot 24:1224–1237

Rolf M, Steil J, Gienger M (2010) Goal babbling permits direct learning of inverse kinematics. IEEE Trans Auton Ment Dev 2:216–229

Salvietti G, Zhang H, Gonzalez-Gomez J, Prattichizzo D, Zhang J (2009) Task priority grasping and locomotion control of modular robot. In: Proceedings of the 2009 IEEE international conference on robotics and biomimetics, Guilin, pp 1069–1074

Sciavicco L, Siciliano B (1988) A solution algorithm to the inverse kinematic problem for redundant manipulators. IEEE J Robot Autom 4:403–410

Whitney D (1969) Resolved motion rate control of manipulators and human prostheses. IEEE Trans Man Mach Syst 10(2):47–53

Regulation and Tracking of Nonlinear Systems

Lorenzo Marconi
C.A.SY. Ű- DEI, University of Bologna, Bologna, Italy

Abstract

A classical problem in control theory is the design of feedback laws such that the effect of exogenous inputs on selected output variables is asymptotically rejected. This includes problems of asymptotic tracking and disturbance rejection. In this entry, the fundamentals of the theory are presented, as well as constructive procedures for the design of a controller, which embeds an "internal model" of the generator of the exogenous inputs. Current and future research directions are also discussed.

Keywords

Introduction

The problem of controlling a dynamical systems in such way that a "regulated" output tracks reference signals or rejects exogenous disturbances is ubiquitous in control theory. Among various possible different approaches to the solution of this problem, in this entry we present the so-called theory of nonlinear output regulation. A distinctive feature of this theory is that reference/disturbance signals to be tracked/rejected are thought of as unknown functions of time, which belong to the set of all trajectories generated by an autonomous nonlinear system (the so-called *exosystem*). Fundamental in this setting is the concept of *internal model*, developed in the early 1970s for linear systems by Francis and Wonham (1976) and subsequently extended, beginning with the work (Isidori and Byrnes 1990), to the case nonlinear systems. Since these early contributions, nonlinear output regulation has been an active research domain, in which constant improvements have brought the theory to a stage of full maturity. In this entry we introduce the fundamental principles of the nonlinear output regulation theory and the associated design tools. The entry ends with an overview of actual research trends and future research directions.

The Generalized Tracking Problem for Nonlinear Systems

We consider the class of time-invariant smooth nonlinear systems described in the form

$$\begin{aligned} \dot{x} &= f(w, x, u) \\ e &= h(w, x) \\ y &= k(w, x) \end{aligned} \qquad (1)$$

in which $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, $y \in \mathbb{R}^q$ is the measured output, and

$e \in \mathbb{R}^p$ is the regulation *error*. The input $w \in \mathbb{R}^s$ models exogenous signals that might represent references to be tracked, exogenous disturbances to be rejected, or also parametric uncertainties. In this framework the problem is to design a controller of the form

$$\begin{aligned}\dot{\xi} &= \varphi(\xi, y) \qquad \xi \in \mathbb{R}^\nu \\ u &= \gamma(\xi, y)\end{aligned} \qquad (2)$$

such that the associated closed-loop system (1)–(2) has bounded trajectories $(x(t), \xi(t))$ and the resulting error $e(t) = h(w(t), x(t))$ is asymptotically vanishing, i.e., $\lim_{t\to\infty} e(t) = 0$. The previous framework encompasses several standard control problems, such as the problem in which a system of the form $\dot{x} = f(x, u)$, with measured output $y = k(x)$ and regulated output $y_r = h_r(x)$, must be controlled in such a way that $y_r(t)$ asymptotically tracks a reference signal $y^*(t)$. This is the case, in fact, if we set $w(t) = y^*(t)$, define $e = h(w, x) = w - h_r(x)$, and drop the dependence from w in the functions $f(\cdot)$ and $k(\cdot)$ in (1). Similarly, the previous framework lends itself to capture a scenario of disturbance suppression, in which, in a system of the form $\dot{x} = f(d, x, u)$, with measured output $y = k(x)$ and regulated output $y_r = h(x)$, the effect of a disturbance $d(t)$ on the regulated output $y_r(t)$ must be asymptotically rejected. This is the case if we set $w(t) = d(t)$ and drop the dependence on w in $h(\cdot)$ and $k(\cdot)$ in (1). Similarly, by letting $h(x) = x$ and interpreting the variable w as parametric uncertainty, the previous setting captures the problem of robust output feedback stabilization, at the origin, of an uncertain system of the form $\dot{x} = f(w, x, u)$ with measured output $y = k(w, x)$. Of course, the general case of tracking reference signals in presence of exogenous disturbances can be cast in a similar manner.

The ability of solving the problem in question strongly depends on the amount of knowledge one assumes about the exogenous variable w in the design of the controller (2). Among the different options available, in this entry we present the so-called theory of output regulation, in which the exogenous variable is assumed to be an *un-*

known member of a *known family* of functions of time. Specifically, it is assumed that $w(t)$ is an unspecified member of the set of all trajectories generated by an autonomous nonlinear system of the form

$$\dot{w} = s(w) \qquad (3)$$

as its initial condition $w(0)$ ranges on a prescribed set $W \subset \mathbb{R}^s$. In this framework, system (3), usually referred to as the "exosystem," is assumed to be known and its knowledge potentially exploitable in the design of (2). The specific "member" $w(t)$ of the family, however, is unspecified as the initial condition $w(0)$ is not known. The fact of regarding $w(t)$ as unknown member of a known family seems to be the right trade-off between the favorable but unrealistic situation in which $w(t)$ is assumed to be perfectly known and the opposite realistic but conservative situation in which $w(t)$ is regarded as a totally unknown signal. An elementary, and yet meaningful, example is given by the case in which $w(t)$ belongs to the family of periodic functions of time with an unspecified frequency, phase, and amplitude. In this case the exosystem (3) is a *nonlinear* system of the form ($w \in \mathbb{R}^3$)

$$\dot{w}_1 = w_2 \qquad \dot{w}_2 = -w_3^2 w_1 \qquad \dot{w}_3 = 0.$$

Solutions of the previous system, in fact, are periodic functions, with frequency $w_3(t) \equiv w_3(0)$ and amplitude and phase depending on the specific initial condition $(w_1(0), w_2(0))$. Other situations, such as exosystems modeling nonlinear oscillators, can be dealt with in a similar fashion.

In the previous context, the problem of output regulation can be formally cast as follows. Let $X \subset \mathbb{R}^n$ be a set of initial conditions for (1). Then, the problem consists in finding a controller of the form (2), with initial conditions in a set $\Xi \subset \mathbb{R}^\nu$, such that the trajectories of the closed-loop system (1)–(2) augmented with (3), originating from an initial condition $(w(0), x(0), \xi(0)) \in W \times X \times \Xi$, are bounded and $\lim_{t\to\infty} e(t) = 0$ *uniformly* in the initial conditions (The property of "uniformity" is relevant in the context of output regulation. It reflects the requirement that the time needed for the error $e(t)$ to reach an

R

ϵ-neighborhood of the origin only depends on the set $W \times X \times -i$, where the initial conditions are supposed to range, and on ϵ but not on the particular value of the initial condition within $W \times X \times \varXi$.). Depending on the assumptions on the set X where the initial conditions of the plant are assumed to range, the problem is further classified in *semiglobal output regulation*, if the set X is a known *compact* but otherwise arbitrary set of $\mathbb{R}^n$, or *global output regulation* if $X = \mathbb{R}^n$.

Output Regulation Principles

Steady State for Nonlinear Systems and Internal Model Principle

Since the objective is to design the controller such that the effect of the exogenous variable is *asymptotically* rejected by the regulation error, it is apparent that any approach to the solution of the problem of output regulation must be necessarily grounded on a precise characterization of the notion of "steady state" for a nonlinear system. As it is the case for the familiar version of this concept in linear systems theory, a notion of "steady state," for the system consisting of (1)–(3), should be able to capture the "limiting behavior" – if any – that such system asymptotically approaches when the "transient behavior," due to the effect of specific initial conditions of plant and controller, fades out and a "persistent behavior," induced only by the specific exogenous input, emerges. In this respect, the mathematical tool that has been shown to be at the core of a rigorous notion of steady state for nonlinear systems is the one of *ω-limit set of a set*. We refer the reader to Hale et al. (2002) for a definition of this notion and to Byrnes and Isidori (2003) for a description of its use in the characterization of the steady-state behavior of a nonlinear system. In this entry, we simply observe that if the trajectories of the system (3)–(1)–(2) that originate from the set of initial conditions $W \times X \times \varXi$ are bounded (which, in turn, is one of the requirements of the problem in question), then there exists a compact set $\mathcal{A} \subset \mathbb{R}^s \times \mathbb{R}^n \times \mathbb{R}^\nu$, which is precisely the ω-limit set of the set $W \times X \times \varXi$ under the dynamics of (3)–(1)–(2), that is *invariant* for the closed-loop system

and that *uniformly* attracts its trajectories. The set $\mathcal{A}$ is usually referred to as *steady-state locus*, while the restriction of the closed-loop dynamics to the set $\mathcal{A}$ are the *steady-state dynamics* of the closed-loop system. The latter characterize the "limiting behavior" of the system towards which all the closed-loop trajectories converge to. Unlike the case of linear systems, though, in a nonlinear context we cannot expect, in general, that the steady-state behavior is only governed by the exogenous w, namely, that the asymptotic behavior of the closed-loop system is totally independent of the initial conditions of the plant and of the regulator. Assuming that the set W is compact and invariant for (3), it can be proven (see Isidori and Byrnes 2008) that the set $\mathcal{A}$ is the graph of a *set-valued* map defined on W, namely, that there exists a map $\sigma : W \to \mathbb{R}^n \times \mathbb{R}^\nu$, which is set-valued in general, such that

$$\mathcal{A} = \{(w, x, \xi) \in W \times \mathbb{R}^n \times \mathbb{R}^\nu \ : \ (x, \xi) \in \sigma(w)\}.$$

Clearly the steady-state locus and the associated steady-state dynamics of the closed-loop system depend on the design of the controller (2). The role of the latter is not only to enforce the existence of a steady state, which is equivalent to enforce bounded closed-loop trajectories, but also to guarantee that the error converges asymptotically to zero uniformly in the initial conditions. In this respect, by bearing in mind the asymptotic properties of the set $\mathcal{A}$, it can be seen that a sufficient condition under which a regulator of the form (2) solves the problem of output regulation is that the steady-state locus is "shaped" in such a way that the regulation error is zero on it, namely,

$$\mathcal{A} \subset \{(w, x, \xi) \ : \ h(w, x) = 0\}. \qquad (4)$$

In fact, it can be proved that condition (4) is not only sufficient but also necessary (see Byrnes and Isidori 2003) as a consequence of the requirement that the error converges to zero uniformly in the initial conditions. That is, any regulator that solves the problem in question necessarily enforces a steady state such that the steady-state locus fulfills (4).

In view of the previous considerations, a crucial property required to any regulator is to induce a steady-state locus $\mathcal{A}$ fulfilling (4). This key feature can be further elaborated by highlighting two necessary conditions, involving separately the plant and the regulator, leading to the notions of *regulator equations* and *internal model property*. To this purpose, consider the simplified, yet relevant, case in which the map $\sigma(\cdot)$ is single-valued and smooth, and let $\pi(w)$ and $\tau(w)$ be the two components of $\sigma(w)$ associated to x and ξ, respectively. By letting

$$c(w) = \gamma(\tau(w), k(w, \pi(w))) \qquad (5)$$

it is immediately realized that the fact that $\mathcal{A}$ is invariant for (1)–(3) implies that the functions $\pi(\cdot)$ and $\tau(\cdot)$ necessarily fulfill

$$\frac{\partial \pi(w)}{\partial w} s(w) = f(w, \pi(w), c(w)) \qquad (6)$$

and

$$\frac{\partial \tau(w)}{\partial w} s(w) = \varphi(\tau(w), k(w, \pi(w))) \qquad (7)$$

for all $w \in W$. Furthermore, the fact that e must be zero on $\mathcal{A}$ (see (4)) implies that necessarily

$$h(w, \pi(w)) = 0 \qquad (8)$$

for all $w \in W$. Equations (6) and (8), interpreted as equations in the unknown $\pi(w)$ and $c(w)$, involve only the regulated plant (1) and are known as *regulator equations* (see Isidori 1995; Isidori and Byrnes 1990). The functions $c(w(t))$ and $\pi(w(t))$, with $w(t)$ solution of (3), represent, respectively, the desired steady-state control input and state towards which the actual control input u and state x of (1) should converge in order to have the regulation goal fulfilled. On the other hand, Eqs. (5) and (7), interpreted as equations in the unknown $\tau(w)$, point out the so-called internal model property required to any regulator solving the output regulation problem, that is, the ability of the regulator to reproduce the ideal steady-state input $c(w(t))$, for all possible $w(t)$ solution of (3), once it is driven by the measured output of

the plant in the ideal steady state (namely, by the function $k(w(t), \pi(w(t)))$). In fact, this property can be achieved by incorporating in the controller an appropriate "internal model" of the exogenous dynamics (3).

Regulator Design

As emphasized in the previous discussion, the design of the regulator involves the fulfillment of two crucial properties. The first is the internal model property, namely, the ability of generating, by means of the regulator outputs, all possible "feedforward inputs" which force an identically zero regulation error and, in turn, to guarantee the existence of an invariant steady-state set on which the error is identically zero. The second property asks that such a steady-state set is asymptotically stable for the closed-loop system with a domain of attraction including the set of initial conditions. A systematic design procedure of regulators simultaneously fulfilling the previous two properties can be found under sufficient conditions that essentially restrict the class of regulated plants (1). In particular, in the following, we consider the class of single input-single output systems that are affine in the input u, with a measurable error variable (i.e., $e = y$) and that after an appropriate change of coordinates can be written in the form

$$\begin{aligned} \dot{z} &= f_z(w, z, e) & z &\in \mathbb{R}^{n-1} \\ \dot{e} &= a(w, z, e) + b(w, z, e)u & e, u &\in \mathbb{R} \end{aligned} \qquad (9)$$

with $f_z(\cdot, \cdot, \cdot)$, $a(\cdot, \cdot, \cdot)$, and $b(\cdot, \cdot, \cdot)$ smooth functions with $b(w, z, e) \neq 0$ for all (w, z, e). Systems of this kind possess a well-defined unitary relative degree (The restriction to systems with unitary relative degree is just made for sake of simplicity. Higher relative degree can be equally dealt with, Isidori (1995).) between the input u and the output e, and the Eqs. (9) are said to be in *normal form* (see Isidori 1995, 2013). In these coordinates an easy calculation shows that the solution of the regulator Eqs. (6) and (8) takes the form $\pi(w) = (\pi_z(w), 0)$, where $\pi_z(\cdot) : W \to \mathbb{R}^{n-1}$ is a solution of

R

$$\frac{\partial \pi_z(w)}{\partial w} s(w) = f_z(w, \pi_z(w), 0),$$

and

$$c(w) = -\frac{a(w, \pi_z(w), 0)}{b(w, \pi_z(w), 0)}.$$

In addition, we further restrict the class of systems by asking for a minimum-phase property (Isidori 1995, 2013). In the present context, the property in question amounts to asking that the set $\mathcal{B} = \{(w, z) \in W \times \mathbb{R}^{n-1} : z = \pi_z(w)\}$ is *asymptotically stable* for the system

$$\dot{w} = s(w)$$
$$\dot{z} = f_z(w, z, 0)$$

with a domain of attraction containing $W \times Z$, with Z the set where the initial condition of z is expected to range.

Existence of the relative degree and the property of minimum-phase are all what is needed to design a regulator. The regulator takes the form

$$\dot{\xi} = F\xi + G(\gamma(\xi) + \kappa(e)) \quad \xi \in \mathbb{R}^\nu$$
$$u = \gamma(\xi) + \kappa(e) \tag{10}$$

in which (F, G) is a controllable pair and $\gamma(\cdot)$ and $\kappa(\cdot)$ are real-valued functions to be properly designed. In particular, it can be shown (see Marconi et al. 2007) that if ν, the dimension of the regulator is taken sufficiently large relative to the dimension of w (specifically, $\nu \geq 2s + 2$) and if F is any matrix whose eigenvalues have negative real part, there exist a continuously differentiable function $\tau(\cdot)$ and a continuous function $\gamma(\cdot)$ such that

$$\frac{\partial \tau(w)}{\partial w} s(w) = F\tau(w) + Gc(w)$$
$$c(w) = \gamma(\tau(w)) \tag{11}$$

for all $w \in W$. This being the case, it is seen that the regulator (10) fulfills conditions (5)–(7) and therefore has the internal model property, regardless of how $\kappa(\cdot)$ is chosen, provided that $\kappa(0) = 0$. In particular, in the closed-loop system (3), (9), and (10), the invariant set $\mathcal{A} = \{(w, z, e, \xi) \in W \times \mathbb{R}^{n-1} \times \mathbb{R} \times \mathbb{R}^\nu : z = \pi_z(w), e = 0, \xi = \tau(w)\}$ fulfills (4) regardless

of how $\kappa(\cdot)$ is chosen. The function $\kappa(\cdot)$ is a degree of freedom that can be chosen to make the steady-state set $\mathcal{A}$ asymptotically stable. In this respect the minimum-phase assumption and the fact that F is a Hurwitz matrix play a role. In fact, the closed-loop system (3), (9), and (10), interpreted as a system with state (w, z, e), input $\kappa(\cdot)$, and output e, have relative degree one and it is minimum-phase. This fact makes it possible to use standard high-gain arguments to show that there exists a function $\kappa(\cdot)$ such that the set $\mathcal{A}$ is asymptotically stable for the closed-loop systems with a domain of attraction containing any (arbitrarily large) compact set of initial conditions (see Marconi et al. 2007; Teel and Praly 1995).

The delicate part in the procedure illustrated above is the design of the function $\gamma(\cdot)$ that is required to fulfill (11) for a suitable $\tau(\cdot)$. Exact, although hardly implementable in practice, expressions for the function $\gamma(\cdot)$ can be found in Marconi and Praly (2008). More constructive design procedures can be found at the price of restricting the class of systems and exosystems that can be dealt with. Such procedures require that the autonomous dynamical system with "output" u^*

$$\dot{w} = s(w)$$
$$u^* = c(w),$$

namely, the system characterizing all possible ideal steady-state inputs, is "immersed" into a system exhibiting certain structural properties (Loosely speaking, the autonomous system with output Σ is immersed into the autonomous system with output $\tilde{\Sigma}$ if the set of all possible functions of time generated as outputs of Σ is a subset of the set of all possible functions generated as outputs of $\tilde{\Sigma}$.). In this respect a number of alternative solutions have been proposed in literature that differ for the kind of underlying immersion assumption and consequent regulator design procedure. Immersion into a linear known observable system (see Byrnes et al. 1997; Huang and Lin 1994; Khalil 1994; Serrani et al. 2000), immersion into a linear unknown (but linearly parameterized) system (Serrani et al. 2001), immersion into a linear system having a nonlinear output map (Chen and Huang 2004), immersion

into a nonlinear system linearizable by output injection (Delli Priscoli 2004), immersion into a system in canonical observability form Byrnes and Isidori (2004), and immersion into a system in a nonlinear adaptive observability form Delli Priscoli et al. (2006a,b) are a few examples of approaches proposed in literature.

Summary and Future Directions

The theory of output regulation for nonlinear systems is an active area of investigation. Research efforts are, in particular, addressed to the problems of weakening the minimum-phase assumption and of identifying *robust* design procedures, to asymptotically stabilize the steady-state locus, not necessarily based on high-gain principles. Recently, the problem of output regulation for multivariable systems has been also addressed (Isidori and Marconi 2012). In this case a paradigm shift in the design of the regulator and of the stabilizer is expected to deal with the problem in its full generality. Finally, it is worth mentioning that the theory of output regulation and internal model-based design methods are being used for the problem of reaching a consensus between the outputs of a network of nonlinear systems exchanging relative information over a communication graph. In this case it has been proved the necessity of internal model-based regulators (Wieland 2010) and the research activity is now conveyed to identify constructive design strategies for classes of nonlinear systems and network topologies.

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Lyapunov's Stability Theory
▶ Nonlinear Zero Dynamics
▶ Tracking and Regulation in Linear Systems

Bibliography

Byrnes CI, Isidori A (2003) Limit sets, zero dynamics and internal models in the problem of nonlinear output regulation. IEEE Trans Autom Control 48:1712–1723

Byrnes CI, Isidori A (2004) Nonlinear internal models for output regulation. IEEE Trans Autom Control 49:2244–2247

Byrnes CI, Delli Priscoli F, Isidori A, Kang W (1997) Structurally stable output regulation of nonlinear systems. Automatica 33:369–385

Chen Z, Huang J (2004) Global robust servomechanism problem of lower triangular systems in the general case. Syst Control Lett 52:209–220

Delli Priscoli F (2004) Output regulation with nonlinear internal models. Syst Control Lett 53: 177–185

Delli Priscoli F, Marconi L, Isidori A (2006a) A new approach to adaptive nonlinear regulation. SIAM J Control Optim 45:829–855

Delli Priscoli F, Marconi L, Isidori A (2006b) Nonlinear observers as nonlinear internal models. Syst Control Lett 55:640–649

Francis BA, Wonham WM (1976) The internal model principle of control theory. Automatica 12:457–465

Hale JK, Magalhães LT, Oliva WM (2002) Dynamics in infinite dimensions. Springer, New York

Huang J (2004) Nonlinear output regulation: theory and applications. SIAM, Philadelphia

Huang J, Lin CF (1994) On a robust nonlinear multivariable servomechanism problem. IEEE Trans Autom Control 39:1510–1513

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, Berlin/New York

Isidori A (2013) Nonlinear zero dynamics. In: Encyclopedia of systems and control. Springer

Isidori A, Byrnes CI (1990) Output regulation of nonlinear systems. IEEE Trans Autom Control 25:131–140

Isidori A, Byrnes CI (2008) Steady-state behaviors in nonlinear systems with an application to robust disturbance rejection. Ann Rev Control 32:1–16

Isidori A, Marconi L (2008) System regulation and design, geometric and algebraic methods'. In: Encyclopedia of complexity and system science. Section control and dynamical systems. Springer, Heidelberg

Isidori A, Marconi L (2012) Shifting the internal model from control input to controlled output in nonlinear output regulation. In: Proceedings of the 51th IEEE conference on decision and control, Maui

Isidori A, Marconi L, Serrani A (2003) Robust autonomous guidance: an internal model-based approach. Limited series advances in industrial control. Springer, London

Khalil H (1994) Robust servomechanism output feedback controllers for feedback linearizable systems. Automatica 30:587–1599

Marconi L, Praly L (2008) Uniform practical output regulation. IEEE Trans Autom Control 53(5):1184–1202

Marconi L, Praly L, Isidori A (2007) Output stabilization via nonlinear luenberger observers. SIAM J Control Optim 45(6):2277–2298

Pavlov A, van de Wouw N, Nijmeijer H (2006) Uniform output regulation of nonlinear systems: a convergent dynamics approach. Birkhauser, Boston

R

Serrani A, Isidori A, Marconi L (2000) Semiglobal output regulation for minimum-phase systems. Int J Robust Nonlinear Control 10:379–396

Serrani A, Isidori A, Marconi L (2001) Semiglobal nonlinear output regulation with adaptive internal model. IEEE Trans Autom Control 46:1178–1194

Teel AR, Praly L (1995) Tools for semiglobal stabilization by partial state and output feedback. SIAM J Control Optim 33:1443–1485

Wieland P (2010) From static to dynamic couplings in consensus and synchronization among identical and non-identical systems. PhD thesis, Universität Stuttgart

Risk-Sensitive Stochastic Control

Hideo Nagai
Osaka University, Osaka, Japan

Abstract

Motivated by understanding "robustness" from the view points of stochastic control, the studies of risk-sensitive control have been developed. The idea was applied to portfolio optimization problems in mathematical finance, from which new kinds of problem on stochastic control, named "large deviation control," have been brought, and currently the studies are in progress.

Keywords

Large deviation control; Mathematical finance; Robustness

Risk-Sensitive Criterion

Risk-sensitive stochastic control has the criterion

$$J(x, 0; T; \gamma) = \frac{1}{\gamma} \log E[e^{\gamma\{\int_0^T f(X_s, u_s)ds + \varphi(X_T)\}}] \tag{1}$$

with $\gamma \neq 0$, where X_t is the state variable process defined by the controlled stochastic differential equation

$$dX_t = \sigma(X_t)dB_t + b(X_t, u_t)dt, \quad X_0 = x \tag{2}$$

with the control parameter process u_t. Here $\sigma(x) : R^N \mapsto R^N \otimes R^d$ and $b(x, u) : R^N \times R^m \mapsto R^N$. When $\gamma \to 0$, the criterion behaves as

$$J(x, 0; T; \gamma) \sim E[\int_0^T f(X_s, u_s)ds + \varphi(X_T)]$$
$$+ \frac{\gamma}{2} \text{Var}[\int_0^T f(X_s, u_s)ds + \varphi(X_T)] + O(\gamma^2).$$

Then, minimizing the criterion with $\gamma > 0$ is considered to be risk averse, while with $\gamma < 0$ it is to be risk seeking. The problem minimizing the classical criterion $E[\int_0^T f(X_s, u_s)ds + \varphi(X_T)]$ corresponds to the case of $\gamma = 0$, which is risk neutral.

When $f(x, u) = \frac{1}{2}x^*Qx + \frac{1}{2}u^*Su$, $\varphi(x) = \frac{1}{2}x^*Ux$, and $b(x, u) = Ax + Cu$, $\sigma(x) \equiv \Sigma$ with constant matrices Q, S, U, A, C, Σ, minimizing the criterion subject to the state variable processes X_t is called a linear exponential quadratic Gaussian (LEQG) control problem, where one may assume Q and U to be nonnegative definite and S positive definite.

H-J-B Equations

The Hamilton-Jacobi-Bellman (H-J-B) equation for the problem minimizing criterion J defined by (1) among the controlled processes governed by (2) is seen to be

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2}\text{tr}[a(x)D^2v] + H(x, \nabla v) = 0 \\ v(T, x) = \varphi(x), \end{cases} \tag{3}$$

where $a(x) := (a^{ij}(x)) = ((\sigma\sigma^*)^{ij}(x))$ and

$$H(x, p) = \frac{\gamma}{2} p^* a(x) p + \inf_u \{b(x, u)^* p + f(x, u)\}.$$

In an LEQG case, where we assume that Q, U are nonnegative definite and S positive definite, the H-J-B equation has the solution expressed as

$$v(t, x) = \frac{1}{2}x^* P(t)x + G(t),$$

by using the solutions $G(t)$ of ordinary differential equation

$$\dot{G}(t) + \frac{1}{2}\text{tr}[P\Sigma\Sigma^*] = 0, \quad G(T) = 0$$

and $P(t)$ of the Riccati equation

$$\dot{P}(t) + PA + A^*P - P(CS^{-1}C^* - \gamma\Sigma\Sigma^*)$$
$$P + Q = 0$$

with the terminal condition $P(T) = U$, provided that it has a nonnegative definite solution $P(t)$. However, it may occur that the Riccati equation does not have any solution if γ is large. In that case, we say that the risk-sensitive control problem "breaks down." Namely, there is no control which makes the criterion have a finite value. On the other hand, if it has a solution, then the optimal feedback control is seen to be $-S^{-1}C^*P(t)x$ and the optimal diffusion process turns out to be the solution to

$$dX_t = \Sigma dB_t + (AX_t - CS^{-1}C^*X_t)dt, \ X_0 = x.$$

The situation can extend to certain general cases. Under sufficiently general conditions one can say that if H-J-B equation (3) has a solution, then no "breakdown" occurs in the corresponding risk-sensitive stochastic control problem (cf. Bensoussan and Nagai 2000; Bensoussan et al. 1998; Nagai 1996).

The LEQG problems were first investigated in Jacobson (1973), and then a theory of the LEQG control with complete or partial state information is developed in Whittle (1981) and Bensoussan and Van Schuppen (1985). Development of the studies of nonlinear risk-sensitive control can be seen in Bensoussan et al. (1998), Nagai (1996), Fleming and McEneaney (1995), etc.

Singular Limits and H^∞ Control

The large deviation theory of Freidlin-Wentzell applies to the risk-sensitive control problem with the criterion

$$J_\epsilon(x, 0; T) = \frac{\epsilon}{\theta}\log E[e^{\frac{\theta}{\epsilon}\int_0^T\{\frac{1}{2}u_s^*S(X_s)u_s + V(X_s)\}ds}]$$
(4)

and the controlled dynamics

$$dX_t = \sqrt{\epsilon}\sigma(X_t)dB_t + \{b(X_t) + C(X_t)u_t\}dt.$$
(5)

The corresponding H-J-B equation is

$$\begin{cases} \frac{\partial v_\epsilon}{\partial t} + \frac{\epsilon}{2}\text{tr}[a(x)D^2v_\epsilon] + H_0(x, \nabla v_\epsilon) + V = 0 \\ v_\epsilon(T, x) = 0, \end{cases}$$
(6)

$$\begin{aligned} H_0(x, p) &= \frac{\theta}{2}p^*a(x)p + b(x)^*p \\ &\quad + \inf_{u\in R^m}\{u^*C(x)p + \frac{1}{2}u^*S(x)u\} \\ &= b(x)^*p - \frac{1}{2}p^*\{CS^{-1}C(x)^* - \theta a(x)\}p. \end{aligned}$$

By employing viscosity solution theory, we can see that, when sending $\epsilon \to 0$, the solution v_ϵ of (6) converges to the viscosity solution w of the equation

$$\begin{cases} \frac{\partial w}{\partial t} + H_0(x, \nabla w) + V = 0 \\ w(T, x) = 0. \end{cases}$$
(7)

Noting that $H_0(x, p)$ can be regarded as

$$\begin{aligned} H_0(x, p) &= \sup_z\{z^*p - \frac{1}{2\theta}z^*a(x)^{-1}z\} \\ &\quad + \inf_u\{u^*C(x)p + \frac{1}{2}u^*S(x)u\}, \end{aligned}$$

Equation (7) is written as

$$\begin{aligned} &\frac{\partial w}{\partial t} + \sup_z\{z^*Dw - \frac{1}{2\theta}(Dw)^*a(x)^{-1}Dw\} \\ &\quad + \inf_u\{u^*C(x)Dw + \frac{1}{2}u^*S(x)u\} = 0 \\ &w(T, x) = 0. \end{aligned}$$

This equation has a unique viscosity solution under suitable conditions. Further, $w(0, x)$ is characterized as the lower value of the differential game with the criterion

$$I(0, T; z_., u(z_.)) = \int_0^T \Psi(x_s, z_s, u(z_.)_s)ds,$$

$$\Psi(x, z, u) = -\frac{1}{2\theta}z^*a(x)^{-1}z + \frac{1}{2}u^*S(x)u + V(x)$$

and the controlled dynamics

$$dx_s = \{b(x_s) + z_s + C(x_s)u(z_.)_s\}ds, \quad x_0 = x,$$

where z_s is a measurable, R^N-valued function on $[0, T]$ such that $\int_0^T |z_s|^2ds < \infty$ and the set of such $\{z_s\}$ is denoted by $\mathcal{Z}_T$. Further, let $\mathcal{U}$ be the

totality of a measurable, R^m-valued function such that $\int_0^T |u_s|^2 ds < \infty$ and $u(z.)$ be a map defined on $\mathcal{Z}$ with its value on $\mathcal{U}$ such that whenever for each $0 \leq \tau \leq T$ and $z^{(1)}, z^{(2)} \in \mathcal{Z}$, $z^{(1)}(s) = z^{(2)}(s)$, almost everywhere on $0 \leq s \leq \tau$, then $u(z.^{(1)})_s = u(z.^{(2)})_s$, a.e. on $0 \leq s \leq \tau$, and the set of such $u(z.)$ is denoted by Γ_U. Thus, the lower value of the game is defined as

$$w(0, x) = \inf_{u(z.) \in \Gamma_U} \sup_{z. \in \mathcal{Z}} I(0, T; z., u(z.))$$

(cf. Bensoussan and Nagai 1997 and references therein). The differential game is known to be related to H^∞ or Robust control. If θ is large, then H-J-B equation (6) may fail to have a solution (cf. Bensoussan and Nagai 1997, 2000). The size of θ ensuring the existence of solution to (6) is related to the level of robustness which the above differential game concerns (Basar and Bernhard 1991; Bensoussan and Nagai 1997, 2000; Bensoussan et al. 1998; Whittle 1990).

Risk-Sensitive Asset Management

The idea of risk-sensitive control applies to mathematical finance (Bielecki and Pliska 1999; Fleming 1995). Consider a market model with $m + 1$ securities, where the security prices are defined by

$$dS^0(t) = r(X_t)S^0(t)dt,$$

$$dS^i(t) = S^i(t)\{\alpha^i(X_t)dt + \sum_{k=1}^{n+m} \sigma_k^i(X_t)dB_t^k\},$$

$i = 1, \ldots m$, with an $n + m$ dimensional Brownian motion process $B_t = (B_t^1, B_t^2, \ldots, B_t^{n+m})$ defined on a filtered probability space $(\Omega, \mathcal{F}, P; \mathcal{F}_t)$. The volatilities σ, the instantaneous mean returns α of the risky assets, and the interest rate r of the riskless asset are affected by the economic factors $(X_t^1, \ldots, X_t^n)$ defined as the solution of the stochastic differential equation

$$dX_t = \beta(X_t)dt + \lambda(X_t)dB_t, \quad X(0) = x \in R^n.$$

Let us set the total wealth W_T of an investor to be $W_T = \sum_i N_T^i S_T^i$ with N_T^i, number of the share invested to ith security S_T^i at time T, and W_0 the initial wealth. Expected power utility maximization maximizing $\frac{1}{\gamma} E[W_T^\gamma] = \frac{1}{\gamma} E[e^{\gamma \log W_T}]$, $\gamma < 1, \neq 0$ (Merton 1990) is equivalent to

$$\sup \frac{1}{\gamma} \log E[e^{\gamma \log W_T}], \tag{8}$$

and it has been studied in terms of "risk-sensitive asset management." When introducing portfolio proportion h_t^i invested to ith security defined by $h^i(t) = \frac{N^i(t)S^i(t)}{W(t)}$ for each $i = 0, \ldots, m$ and setting $h(t)^* = (h^1(t), h^2(t), \ldots, h^m(t))$, the total wealth W_t turns out to satisfy

$$\frac{dW(t)}{W(t)} = \{r(X_t) + h(t)^*\hat{\alpha}(X_t)\}dt$$
$$+ h(t)^*\sigma(X_t)dB_t,$$

under the self-financing condition, where $\hat{\alpha}(x) = \alpha(x) - r(x)\mathbf{1}$, $\mathbf{1} = (1, 1, \ldots, 1)^*$. In considering the maximization problem, the portfolio proportion h_t is considered an investment strategy to be controlled and assumed to be $\mathcal{G}_t^{S,X} := \sigma(S(u), X(u), u \leq t)$ progressively measurable in the case of full information. The problem is often considered under partial information where h_t is assumed to be $\mathcal{G}_t^S := \sigma(S(u), u \leq t)$ measurable. Here we first discuss the case of full information, and the set of admissible strategies $\mathcal{A}(T)$ (or $\mathcal{A}$) is determined as the totality of $\mathcal{G}_t^{S,X}$ progressively measurable investment strategies satisfying some suitably defined integrability conditions.

Considering (8) for $\gamma < 0$ amounts to studying the minimization problem

$$\hat{v}(0, x) = \inf_{h. \in \mathcal{A}(T)} \log E[e^{\gamma \log W_T(h)}]. \tag{9}$$

Then introducing a probability measure p^h defined by

$$P^h(A) = E[e^{\gamma \int_0^T h_s^* \sigma(X_s) dW_s - \frac{\gamma^2}{2} \int_0^T h_s^* \sigma \sigma^*(X_s) h_s ds} : A],$$

$A \in \mathcal{F}_T$, the value function is expressed as

$$\hat{v}(0, x) = \gamma \log W_0 + \inf_{h. \in \mathcal{A}(T)}$$

$$\log E^h [e^{-\gamma \int_0^T \eta(X_s, h_s) ds}]$$

with the initial wealth W_0, where

$$\eta(x, h) = -h^* \hat{\alpha}(x) + \frac{1-\gamma}{2} h^* \sigma \sigma^*(x) h - r(x)$$

and $\hat{\alpha}(x) = \alpha(x) - r(x)\mathbf{1}$. By using the Brownian motion $B_t^h := B_t - \gamma \int_0^t \sigma^*(X_s) h_s ds$ under the new probability measure P^h, the dynamics of the economic factor X_t is written as

$$dX_t = \{\beta(X_t) + \gamma \lambda \sigma^*(X_t) h_t\} dt + \lambda(X_t) dB_t^h. \tag{10}$$

Thus, we arrive at the risk-sensitive control problem with the value function $\hat{v}(0, x)$ and the controlled dynamics X_t governed by (10). Note that $\frac{1-\gamma}{2} > 0$ for $\gamma < 1$ and that the case where $\gamma < 0$ is called risk averse, which we mainly discuss here. Then the corresponding H-J-B equation is deduced as

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2} \text{tr}[\lambda \lambda^* D^2 v] + \frac{1}{2}(Dv)^* \lambda \lambda^* Dv \\ + \inf_h \{[\beta + \gamma \lambda \sigma^* h]^* Dv - \gamma \eta(x, h)\} = 0, \\ v(T, x) = \gamma \log W_0, \end{cases} \tag{11}$$

which can be rewritten as

$$\begin{cases} \frac{\partial v}{\partial t} + \frac{1}{2} \text{tr}[\lambda \lambda^* D^2 v] + \beta_\gamma^* Dv \\ + \frac{1}{2}(Dv)^* \lambda N_\gamma^{-1} \lambda^* Dv - U_\gamma = 0, \\ v(t, x) = \gamma \log W_0. \end{cases} \tag{12}$$

Here $U_\gamma = -\frac{\gamma}{2(1-\gamma)} \hat{\alpha}^* (\sigma \sigma^*)^{-1} \hat{\alpha} + r(x)$, $\beta_\gamma = \beta + \frac{\gamma}{1-\gamma} \lambda \sigma^* (\sigma \sigma^*)^{-1} \hat{\alpha}$ and $N_\gamma^{-1} = I + \frac{\gamma}{1-\gamma} \sigma^* (\sigma \sigma^*)^{-1} \sigma$. Under suitable conditions H-J-B equation (12) has a solution with sufficient regularities (Bensoussan et al. 1998; Nagai 2003). Moreover, identification

$$v(0, x; T) \equiv v(0, x) = \hat{v}(0, x) \tag{13}$$

can be verified. Further, $\hat{h}(t, X_t) = \frac{1}{1-\gamma} (\sigma \sigma^*)^{-1} \{\hat{\alpha}(X_t) + \sigma \lambda^* Dv(t, X_t)\}$ is the optimal investment strategy for problem (9) (Nagai 2003).

A typical example is the case of linear Gaussian model such that $r(x) = r$, $\alpha(x) = Ax + a$, $\sigma(x) = \Sigma$, $\beta(x) = Bx + b$, $\lambda(x) = \Lambda$, where A, B, Σ, Λ are constant matrices; a, b are constant vectors; and r is a constant. Then, the solution to (12) has an explicit representation as $v(t, x) = \frac{1}{2} x^* P(t) x + q(t)^* x + k(t)$, where $P(t)$ is the negative semi-definite solution to the Riccati equation

$$\dot{P}(t) + P(t) \Lambda N^{-1} \Lambda^* P(t) + K_1^* P(t) + P(t) K_1 + \frac{\gamma}{1-\gamma} A^* (\Sigma \Sigma^*)^{-1} A = 0, \quad P(T) = 0 \tag{14}$$

and $q(t)$, $k(t)$ are, respectively, the solutions to

$$\dot{q}(t) + (K_1 + \Lambda N^{-1} \Lambda P(t))^* q(t) + P(t) b + \frac{\gamma}{1-\gamma} (A^* + P(t) \Lambda \Sigma^*) (\Sigma \Sigma^*)^{-1} \hat{a} = 0, \quad q(T) = 0 \tag{15}$$

and

$$\begin{aligned} &\dot{k}(t) + \frac{1}{2} \text{tr}[\Lambda \Lambda^* P(t)] + \frac{1}{2} q(t)^* \Lambda \Lambda^* q(t) \\ &+ \frac{\gamma}{2(1-\gamma)} (\hat{a} + \Sigma \Lambda^* q(t))^* (\Sigma \Sigma^*)^{-1} \\ &(\hat{a} + \Sigma \Lambda^* q(t)) = 0, \\ &k(T) = \gamma \log W_0, \end{aligned} \tag{16}$$

where $K_1 := B + \frac{\gamma}{1-\gamma} \Lambda \Sigma^* (\Sigma \Sigma^*)^{-1} A$, $\hat{a} = a - r\mathbf{1}$ and $N^{-1} := I + \frac{\gamma}{1-\gamma} \Sigma^* (\Sigma \Sigma^*)^{-1} \Sigma$. In this case the optimal strategy has a more explicit form: $\hat{h}_t = \frac{1}{1-\gamma} (\Sigma \Sigma^*)^{-1} [\hat{a} + AX_t] + \frac{1}{1-\gamma} (\Sigma \Sigma^*)^{-1} [\Sigma \Lambda^* q(t) + \Sigma \Lambda^* P(t) X_t]$ (cf. Davis and Lleo 2008; Kuroda and Nagai 2002).

The economic factor X_t may be more suitably considered to be unobservable and then the problem should be formulated as the risk-sensitive stochastic control problem under partial information. Indeed, one can formulate the problem by regarding the log prices $Y_t^i := \log S_t^i$, $i = 0, 1, 2, \ldots, m$ as the observable quantities and the economic factor X_t as the unobservable system process. As for linear Gaussian models and hidden Markov models, the problems are reduced to the ones of full information by obtaining the relevant controlled dynamics in a finite dimension through deducing the filtering equation by

the methods of change of measure (Nagai 1999; Nagai and Runggaldier 2008). Further, one can obtain the explicit form of the optimal strategy, which is $\mathcal{G}_t^S$ measurable, in the case of linear Gaussian model (Nagai 1999) as the parallel result to the above.

Linear Gaussian models for $0 < \gamma < 1$ are extensively studied in Fleming and Sheu (1999, 2002). In that case, one concerns the problems

$$\sup_h \log E[e^{\gamma \log W_T(h)}], \qquad (17)$$

or

$$\bar{\chi}(\gamma) = \sup_h \overline{\lim_{T \to \infty}} \frac{1}{T} \log E[e^{\gamma \log W_T(h)}]. \qquad (18)$$

If $0 < \gamma$ is small, there is a stationary solution of (14) and the verification theorem holds for the problem on infinite horizon (so does for the problem on a finite time horizon). Further, under some conditions there is a threshold $\bar{\gamma}$ such that $\bar{\chi}(\gamma) = \infty$ for $\bar{\gamma} < \gamma$. To know explicitly the size of $\bar{\gamma}$ is important, while it is limited to the case of 1 dimension to be able to realize.

Problems on Infinite Horizon

The value for the problem on infinite time horizon counterpart of (9) is defined as

$$\hat{\chi}(\gamma) = \inf_{h. \in \mathcal{A}} \chi(h; \gamma), \qquad (19)$$

$$\chi(h; \gamma) = \lim_{T \to \infty} \frac{1}{T} \log E[e^{\gamma \log W_T(h)}]$$

when suitably setting the set $\mathcal{A}$ of admissible strategies. The corresponding H-J-B equation of ergodic type for the problem is seen to be

$$\begin{aligned} \chi(\gamma) = &\tfrac{1}{2}\mathrm{tr}[\lambda\lambda^* D^2 w] + \beta_\gamma^* Dw \\ &+ \tfrac{1}{2}(Dw)^*\lambda N_\gamma^{-1}\lambda^* Dw - U_\gamma. \end{aligned} \qquad (20)$$

However, when setting as $\mathcal{A} = \{h.|_{[0,T]} \in \mathcal{A}(T), \quad \forall T\}$, identification of $\hat{\chi}(\gamma)$ with the solution $\chi(\gamma)$ to the H-J-B equation (20) cannot be seen in general. Indeed, even in the case of

linear Gaussian model, such identification does not always hold (Fleming and Sheu 1999; Kuroda and Nagai 2002; Nagai 2003) if $\gamma < 0$. Instead, introduce the asymptotic value

$$\tilde{\chi}(\gamma) = \lim_{T \to \infty} \frac{1}{T} \hat{v}(0, x; T).$$

Then we can see that $\chi(\gamma) = \tilde{\chi}(\gamma)$ under sufficiently general conditions (cf. Hata et al. 2010; Nagai 2012).

In the case of the linear Gaussian model, the solution to the H-J-B equation of ergodic type is given by $w(x) = \frac{1}{2}x^* \bar{P} x + \bar{q}^* x$ with the stationary solutions $\bar{P}$ of (14) and $\bar{q}$ of (15), and if

$$\overline{P}\lambda\Sigma^*(\Sigma\Sigma^*)^{-1}\Sigma\Lambda^*\overline{P} < A^*(\Sigma\Sigma^*)^{-1}A$$

holds, then one can see that $\chi(\gamma) = \hat{\chi}(\gamma)$ (Kuroda and Nagai 2002). Further, the optimal strategy is given by $\hat{h}_t = \hat{h}(X_t)$, with $\hat{h}(x) = \frac{1}{1-\gamma}(\Sigma\Sigma^*)^{-1}[\hat{a} + \Sigma\Lambda^*\bar{q} + (A + \Sigma\Lambda^*\overline{P})x]$ (Kuroda and Nagai 2002). Decomposition as $\hat{h}_t = \frac{1}{1-\gamma}\hat{h}_t^1 + \frac{1}{1-\gamma}\hat{h}_t^2 := \frac{1}{1-\gamma}(\Sigma\Sigma^*)^{-1}[\hat{a} + AX_t] + \frac{1}{1-\gamma}(\Sigma\Sigma^*)^{-1}[\Sigma\Lambda^*\bar{q} + \Sigma\Lambda^*\overline{P}X_t]$ is regarded as a generalization of Merton's Mutual Funds Theorem (Davis and Lleo 2008; Merton 1990). Here $\hat{h}_t^1$ is a log utility portfolio (Kelly portfolio) (Kelly 1956). See also Nagai and Peng (2002) concerning the partial information counterparts of the results in Kuroda and Nagai (2002).

In relation to the above problems on mathematical finance, a new kind of problem studying

$$I(\kappa) = \lim_{T \to \infty} \frac{1}{T} \inf_{h \in \mathcal{A}(T)} \log P(\log W_T(h) \le \kappa T) \qquad (21)$$

for a given constant κ, arises, and it is called "downside risk minimization." The problem is considered "large deviation control" and can be discussed as the dual to risk-sensitive asset management (19) in the risk-averse case $\gamma < 0$ (Hata et al. 2010; Nagai 2011, 2012). Indeed, we obtain

$$I(\kappa) = -\inf_{k \in (-\infty, \kappa]} \sup_{\gamma < 0} \{\gamma k - \hat{\chi}(\gamma)\}.$$

Further, an asymptotically optimal strategy is given as follows. For given κ, take $\gamma(\kappa)$ which attains the supremum in $\sup_{\gamma < 0}\{\gamma \kappa - \hat{\chi}(\gamma)\}$, then the optimal strategy $\hat{h}(t, X_t), 0 \leq t \leq T$ for problem (9) with $\gamma = \gamma(\kappa)$ forms the asymptotically optimal strategy for (21). Historically, the studies of "upside maximization" concerning

$$\bar{I}(\kappa) = \sup_{h \in \mathcal{A}} \overline{\lim_{T \to \infty}} \frac{1}{T} \log P(\log W_T(h) \geq \kappa T)$$

have been preceding (cf. Pham 2003), and the duality relationship between this and (18) was discussed. To develop further studies for the problem, there are difficulties to know the size of $\bar{\gamma}$ (Cf. Fleming and Sheu 1999, 2002).

Cross-References

▶ Credit Risk Modeling
▶ Financial Markets Modeling
▶ Investment-Consumption Modeling
▶ Option Games: The Interface Between Optimal Stopping and Game Theory

Bibliography

Basar T, Bernhard P (1991) H^{∞} – optimal control and related minimax design problems. Birkhäuger, Boston/Cambridge

Bensoussan A (1992) Stochastic control of partially observable systems. Cambridge University Press, Cambridge

Bensoussan A, Nagai H (1997) Min–max characterization of a small noise limit on risk-sensitive control. SIAM J Control Optim 35:1093–1115

Bensoussan A, Nagai H (2000) Conditions for no breakdown and Bellman equations of risk-sensitive control. Appl Math Optim 42:91–101

Bensoussan A, Van Schuppen JH (1985) Optimal control of partially observable stochastic systems with an exponential-of-integral performance index. SIAM J Control Optim 23:599–613

Bensoussan A, Frehse J, Nagai H (1998) Some results on risk-sensitive control with full information. Appl Math Optim 37:1–41

Bielecki TR, Pliska SR (1999) Risk sensitive dynamic asset management. Appl Math Optim 39:337–360

Davis M, Lleo S (2008) Risk-sensitive benchmarked asset management. Quant Financ 8:415–426

Fleming WH (1995) Optimal investment models and risk-sensitive stochastic control. IMA vol Math Appl 65:75–88

Fleming WH, McEneaney WM (1995) Risk-sensitive control on an infinite horizon. SIAM J Control Optim 33:1881–1915

Fleming WH, Sheu SJ (1999) Optimal long term growth rate of expected utility of wealth. Ann Appl Probab 9(3):871–903

Fleming WH, Sheu SJ (2002) Risk-sensitive control and an optimal investment model. II. Ann Appl Probab 12(2):730–767

Hata H, Nagai H, Sheu SJ (2010) Asymptotics of the probability minimizing a "down-side" risk. Ann Appl Probab 20:52–89

Jacobson DH (1973) Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. IEEE Trans Autom Control 18:124–131

Kelly J (1956) A new interpretation of information rate. Bell Syst Tech J 35:917–926

Kuroda K, Nagai H (2002) Risk sensitive portfolio optimization on infinite time horizon. Stoch Stoch Rep 73:309–331

Merton RC (1990) Continuous time finance. Blackwell, Malden

Nagai H (1996) Bellman equations of risk-sensitive control. SIAM J Cont Optim 34:74–101

Nagai H (1999) Risk-sensitive dynamic asset management with partial information. In: "Stochastics in finite and infinite dimensions", a volume in honor of G. Kallianpur. Birkhäuser, Boston, pp 321–340

Nagai H (2003) Optimal strategies for risk-sensitive portfolio optimization problems for general factor models. SIAM J Control Optim 41:1779–1800

Nagai H (2011) Asymptotics of the probability minimizing a "down-side" risk under partial information. Quant Financ 11:789–803

Nagai H (2012) Downside risk minimization via a large deviation approach. Ann Appl Probab 22:608–669

Nagai H, Peng S (2002) Risk-sensitive dynamic portfolio optimization with partial information on infinite time horizon. Ann Appl Probab 12(1):173–195

Nagai H, Runggaldier WJ (2008) PDE approach to utility maximization for market models with hidden Markov factors. In: Dalang et al (ed) Seminar on stochastic analysis, random fields and applications V. Progress in probability. Birkhäser, Basel, pp 493–506

Pham H (2003) A large deviations approach to optimal long term investment. Financ Stoch 7: 169–195

Whittle P (1981) Risk-sensitive linear/quadratic/Gaussian control. Adv Appl Probab 13:764–767

Whittle P (1990) A risk-sensitive maximum principle. Syst Control Lett 15:183–192

Robot Grasp Control

Domenico Prattichizzo
University of Siena, Siena, Italy

Abstract

Robotic grasping is the process of establishing
a physical connection between the robot (or an
appendage of the robot called the gripper) and
an external object in such a way that the robot
can exert forces and torques on the object. Grasp
control requires the satisfaction of contact con-
straints, of which two types are considered. Form
constraints specify geometric configurations of
the gripper that bring it into contact with the
object to be grasped. This article is principally
concerned with force constraints and force clo-
sure that specify forces exerted on the object
that are sufficient to lift, move, or otherwise
manipulate it.

Keywords

Force constraints; Force closure; Grasp con-
straints; Grasp matrix; Hand Jacobian; Twists;
Wrenches

Introduction

Grasp control refers to the art of controlling the
motion of an object by constraining its dynamics
through contacts with a hand. The process of con-
trolling the grasp is not limited to robotic hands
only but also applies to human hands (Johansson
and Edin 1991) and to all other mechanisms using
contact constraints to control the motion of the
manipulated object (Brost and Goldberg 1996).

A crucial role in the control of grasping is
played by contact constraints. All the interactions
between the robotic hand and the grasped object
occur at the contacts whose understanding is
paramount (Salisbury and Roth 1983). The uni-
lateral nature of contact interaction in grasping

makes the control problems much more challeng-
ing than cooperative manipulation where multiple
arms hold the object rigidly allowing bilateral
force transmission at each contact point (Chiac-
chio et al. 1991).

The importance of unilateral contact con-
straints in grasping led a large part of the
literature to focus on the closure properties
of the grasp (Bicchi 1995). Those properties
refer to the ability of a grasp to prevent
motions of the grasped object relying only
on unilateral frictionless constraints in case
of form closure (Reuleaux 1876) and on
contact constraints with friction in case of force
closure (Nguyen 1988). While form closure is
a purely geometric property of the grasp and
depends on where the unilateral contact points
are on the object, force closure depends on the
ability that the robotic hand has to resist and
apply forces to the object through the contacts
while satisfying the friction constraints. In other
terms force closure directly involves the control
of the robotic hand kinematics and not only the
geometry of the contacts (Bicchi 1995). This
entry focuses on force-closed grasps.

The optimal choice of the contact points on the
object surface is a critical issue known as grasp
planning. Among the many optimal criteria that
have been proposed in the literature to choose the
contact points, I want to recall the one proposed
in Ferrari and Canny (1992) where the grasp-
ing configuration is evaluated according to the
magnitude of the largest worst-case disturbance
wrench that can be resisted by the grasp.

Many approaches have been studied in the
literature on grasp planning in the presence of
uncertainties. The uncertainty can be either due to
the shape of the object which is partially known
or partially sensed as in Goldfeder et al. (2009)
or due to the errors in positioning the fingers on
the object during the grasping (Roa and Suarez
2009). In what follows all the parameters of the
grasp including those related to the hand, the
object, and the contact points are assumed to be
known with no uncertainties.

The main objective of grasp control is that of
tracking a desired trajectory with the grasped
object by applying a set of contact forces

satisfying the friction constraints (Bicchi and Kumar 2000). Complex in-hand object motions can be obtained by rolling and sliding the contact points on the object surface as proposed in Montana (1988) or by using finger gaiting to get large-scale motions (Han and Trinkle 1998). This entry deals with non-rolling and non-sliding contact points and summarizes the fundamental theory of computed-torque control for object trajectory and internal force control proposed in Li et al. (1989). For a comprehensive review of the theory of grasping and its control, the reader is referred to Murray et al. (1994), Shimoga (1996), Okamura et al. (2000), Bicchi and Kumar (2000), and Prattichizzo and Trinkle (2008).

Contact and Grasp Model

Notations and definitions on grasping are taken from Prattichizzo and Trinkle (2008). Refer to Fig. 1 and let $\{N\}$ represent the inertial frame fixed to the palm of the robotic hand. Let $u = [p^T, \phi^T]^T \in \mathbb{R}^6$ denote the vector describing the position and orientation of frame $\{B\}$, fixed to the object, relative to $\{N\}$. Vector ϕ expresses the

Euler angles, the pitch-roll-yaw variables, or the exponential coordinates parameterizing $SO(3)$. Denote by $\nu = [v^T \; \omega^T]^T \in \mathbb{R}^6$ the twist of the object. It is worth to note that ν is not equal to $\dot{u}$, but satisfies $\nu = U(u)\dot{u}$ where matrix $U \in \mathbb{R}^{6\times6}$ is such that UU^T is the identity matrix, and the dot over the variable implies differentiation with respect to time (Murray et al. 1994). The joint variables of the robotic hand are defined by $q = [q_1 \; \ldots \; q_{n_q}]^T \in \mathbb{R}^{n_q}$. Let n_c be the number of contact points. The position of contact point i in $\{N\}$ is defined by the vector $c_i \in \mathbb{R}^3$, in the contact point frame $\{C\}_i$ whose axes are $\{\hat{n}_i, \; \hat{t}_i, \; \hat{o}_i\}$ where the unit vector $\hat{n}_i$ is normal to the tangent plane at the contact, and directed toward the object while the other two unit vectors are orthogonal and lie in the tangent plane.

Two matrices are of utmost importance in grasp analysis: the *grasp matrix G* and the *hand Jacobian J*. These two matrices are computed using the complete grasp matrix, the complete Jacobian, and the contact selection matrix that are defined as follows: the transpose of the *complete grasp matrix* $\tilde{G}^T \in \mathbb{R}^{6n_c \times 6}$ maps the object twist to the n_c twist vectors of the contact frames $\{C\}_i$ as thought on the object $v_{c,\text{obj}} = \tilde{G}^T \nu$, while

Robot Grasp Control,
Fig. 1 A two-fingered hand grasping an object
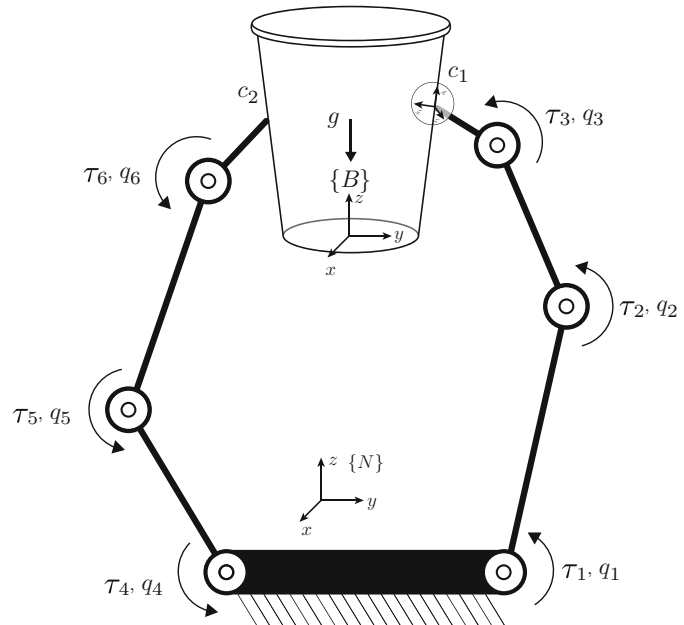
the *complete hand Jacobian Matrix* $\tilde{J} \in \mathbb{R}^{6n_c \times n_q}$ maps the joint velocities to the twists of the contact frames as thought on the hand $v_{c,\mathrm{hnd}} = \tilde{J}\dot{q}$.

When a contact occurs between the hand and the object, assuming no sliding, some components of the relative contact twist between the object and the hand are set to zero according to the used contact model. In this entry the *hard-finger* (HF) and the *soft-finger* (SF) contact models are considered (Mason and Salisbury 1985). Those components are selected by the *contact selection matrix* which selects m components of the relative contact twists for all the contacts and sets them to zero: $H(v_{c,\mathrm{hnd}} - v_{c,\mathrm{obj}}) = 0$. For more details on how to compute the contact selection matrix, the reader is referred to Prattichizzo and Trinkle (2008). Then the following contact constraint equation is obtained:

$$\begin{bmatrix} J & -G^T \end{bmatrix} \begin{bmatrix} \dot{q} \\ v \end{bmatrix} = 0 \qquad (1)$$

where the transpose of the grasp matrix and the hand Jacobian are finally defined by multiplying the contact selection matrix and the transpose of the complete grasp matrix and the complete hand Jacobian as

$$G^T = H\tilde{G}^T \in \mathbb{R}^{m \times 6}$$
$$J = H\tilde{J} \quad \in \mathbb{R}^{m \times n_q}$$

In the force domain, the wrenches that the hand applies to the object at the contact points are collected in the vector λ. Correspondingly, on the hand, a force vector $-\lambda$, opposite to the preceding one, is applied by the object through the contact points. At each contact point, the contact wrenches have components only along the directions constrained by the contact model. Furthermore, contact force components must satisfy the friction constraints (see section "Force Closure and Grasp Control"). More specifically, the m-dimensional vector $\lambda = [\lambda_1^T \ \ldots \ \lambda_{n_c}^T]^T$ contains the contact wrench components applied to the object through the n_c contacts, where the wrench at contact i is defined, for the different contact models here considered, as $\lambda_i = [f_{in} \ f_{it} \ f_{io}]^T$ for the HF contact model and

$\lambda_i = [f_{in} \ f_{it} \ f_{io} \ m_{in}]^T$ for the SF contact model. The subscripts indicate one normal (n) and two tangential (t, o) components of contact force f_i and moment m_i at contact i.

In terms of forces, the grasp matrix maps the transmitted contact wrenches λ to the set of wrenches that the hand can apply to the object $G\lambda$, and the transpose of hand Jacobian maps the contact forces $-\lambda$ to the corresponding vector of joint loads $-J^T\lambda$.

Grouping all the noncontact wrenches applied to the object in $g \in \mathbb{R}^6$ and all the noncontact contributions to the joint loads of the robotic hand in $\tau \in \mathbb{R}^{n_q}$, the rigid-body dynamic equations of the whole system, consisting of the hand and of the grasped object, are

$$M_{\mathrm{obj}}(u)\dot{v} + N_{\mathrm{obj}}(u, v) = G\lambda + g$$
$$M_{\mathrm{hnd}}(q)\ddot{q} + N_{\mathrm{hnd}}(q, \dot{q}) = -J^T\lambda + \tau$$

where $M_{\mathrm{obj}}(\cdot)$ and $M_{\mathrm{hnd}}(\cdot)$ are symmetric, positive definite inertia matrices and $N_{\mathrm{obj}}(\cdot, \cdot)$ and $N_{\mathrm{hnd}}(\cdot, \cdot)$ are the velocity-product terms for the object and the hand, respectively. For the sake of simplicity, the gravity terms are disregarded.

The dynamics of the hand and object are not independent but depend on the kinematic constraints imposed by the contact model (1)

$$\begin{bmatrix} -G \\ J^T \end{bmatrix} \lambda = \begin{bmatrix} \overline{g} \\ \overline{\tau} \end{bmatrix} \qquad (2)$$

$$\text{subject to} \quad J\dot{q} = G^T v$$

where

$$\overline{g} = g - M_{\mathrm{obj}}(u)\dot{v} - N_{\mathrm{obj}}(u, v).$$
$$\overline{\tau} = \tau - M_{\mathrm{hnd}}(q)\ddot{q} - N_{\mathrm{hnd}}(q, \dot{q})$$

It is worth underlying that dynamics can be disregarded for slow motions of the hand and of the object, while it becomes very relevant in applications with high-speed grasping and manipulation as discussed in Namiki et al. (2003).

Controllable Wrenches and Twists

From the first equation in (2) to impose any motion to the object by contact forces, the grasp matrix G must be full row rank, i.e., $\text{rank}(G) = 6$, which is equivalent to have a trivial null space of G^T, i.e., $\mathcal{N}(G^T) = 0$. This is an important property of the grasp which has been referred to as *non-indeterminate* in Prattichizzo and Trinkle (2008) to reflect the idea that the contacts on the object are placed in a way that there are no twists of the object that are not controllable by contact wrenches.

However, this condition depends only on the contacts on the object and does not consider the role of the hand kinematics which comes from the second equation in (2) and from the contact constraint. Under the simplifying assumption that $\mathcal{N}(J^T) = 0$, referred to as *non-defective* grasp in Prattichizzo and Trinkle (2008), it is simple to verify that, for any given contact wrench λ, a control torque τ exists which is able to apply the given contact wrench. The mechanical interpretation of this assumption is that when $\mathcal{N}(J^T) = 0$, there are no contact forces resisted by the robotic hand constraints, i.e., with zero joint load. The simplifying assumption of non-defective grasps ensures that $\mathcal{N}(J^T) \cap \mathcal{N}(G) = 0$ which is a necessary condition to determine the contact force λ from the rigid-body equation (2) as shown in Prattichizzo and Trinkle (2008).

If a grasp is non-defective, it means that each finger of the robotic hand involved in the contact with the object must have a number of joints sufficient to control all the components of the contact wrench. For example, in the case of two HF contact points occurring at the fingertips of a two-fingered robotic hand, each finger must have at least three joints and must be in a non-singular configuration.

This entry does not consider whole-hand or power grasps which, differently from the fingertip grasps, exploit the whole surface of the fingers, including the palm, to constraint the object. The analysis of controllable wrenches and twists for whole-arm grasps, taking into account the hand and object dynamics, can be found in Prattichizzo and Bicchi (1997).

Force Closure and Grasp Control

The dynamic formulation of the grasp with the contact kinematic constraints given in (2) holds only if the contact forces satisfy the friction law imposing constraints on the components of the contact force and moment. Limiting the analysis to HF contact models, Coulomb friction law requires that the components of contact force λ_i at the i-th contact lie inside the friction cone $\mathcal{F}_i$

$$\mathcal{F}_i = \{(f_{in}, f_{it}, f_{io}) \mid \sqrt{f_{it}^2 + f_{io}^2} \le \mu_i f_{in}\} \tag{3}$$

where μ_i represents the friction coefficient at the i-th contact. Extending to all contact points, λ is constrained to lie in $\mathcal{F}$ where $\mathcal{F}$ is the generalized friction cone defined as: $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_{n_c} = \{\lambda \in \mathbb{R}^m \mid \lambda_i \in \mathcal{F}_i; \ i = 1, \dots, n_c\}$.

While grasping an object, the applied contact forces must be consistent with the friction constraints. This is not straightforward for the grasp control and requires to exploit the beneficial characteristics of the internal forces. From the object dynamics in (2), for a given $\overline{g}$, one gets

$$\lambda = -G^+ \overline{g} + N(G)\gamma \tag{4}$$

where G^+ denotes the generalized inverse of the grasp matrix and $N(G)$ denotes a matrix whose columns form a basis for $\mathcal{N}(G)$, and γ is a vector parameterizing the solution set. The contact force λ consists of a particular solution balancing the $\overline{g}$ term and of a homogeneous solution belonging to the null space of the grasp matrix.

In general, the particular solution $-G^+ \overline{g}$ does not satisfy the friction constraint (3) at all the contact points and needs the homogeneous solution $\lambda_h = N(G)\gamma$ to keep the contact forces within the friction cones. Contact forces λ_h in $\mathcal{N}(G)$ are referred to as *internal forces* since they do not contribute to the object dynamics, i.e., $G\lambda_h = 0$. Instead, these forces affect the tightness of the grasp and play a crucial role in maintaining grasps that rely on friction. The existence of a nontrivial null space of the grasp matrix is a

R

desirable property and has been referred to as *graspability* (Prattichizzo and Trinkle 2008).

Another relevant and desirable property of the grasp is the *frictional force closure* which means that for any noncontact wrench $\overline{g}$, an internal force λ_h exists such that the contact force λ in (4) belongs to the generalized friction cone $\mathcal{F}$. In Murray et al. (1994) the authors state that a grasp has frictional form closure if and only if the grasp matrix is full row rank (non-indeterminate grasp) and there exists λ_h such that $G\lambda_h = 0$ and λ_h belong to the interior of the generalized friction cone $\mathcal{F}$.

Grasp control is about using contact forces, which must satisfy the friction constraints, so as to let the object to track a given trajectory. This is also referred to as dexterous manipulation (Bicchi and Kumar 2000). In Li et al. (1989), a computed-torque controller is proposed to track both the desired trajectory of the grasped object u_{des} and the desired internal force $\lambda_{h,\text{des}}$. Under the additional simplifying assumption that the robotic hand Jacobian is invertible, i.e., there are no redundant motions of the fingers, the computed-torque control law

$$
\begin{aligned}
\tau = & N_{\text{hnd}}(q,\dot{q}) + J^T G^+ N_{\text{obj}}(u,v) \\
& - M_{\text{hnd}} J(q) J^{-1} \dot{J} \dot{q} + M_{\text{ho}} \dot{U} \dot{u} \\
& + M_{\text{ho}} U(\ddot{u}_{\text{des}} - K_v \dot{e}_u - K_u e_u) \\
& + J^T(\lambda_{h,\text{des}} - K_s \int e_{\lambda_h}),
\end{aligned}
$$

with $M_{\text{ho}} = M_{\text{hnd}} J(q) J^{-1} G^T + J^T G^+ M_{\text{obj}} J$ guarantees that both the trajectory and the internal force errors

$$
e_u = u - u_{\text{des}}
$$

$$
e_{\lambda_h} = \lambda_h - \lambda_{h,\text{des}}
$$

with respect to the desired object trajectory u_{des} and internal force $\lambda_{h,\text{des}}$ converge to zero according to a second- and first-order dynamics, respectively.

$$
\ddot{e}_u + K_v \dot{e}_u + K_u e_u = 0
$$
$$
e_{\lambda_h} + K_s \int e_{\lambda_h} = 0
$$

where K_v, K_u, and K_s are positive definite matrices.

The computed-torque controller proposed in Li et al. (1989) guarantees only that the desired object trajectory and the desired internal forces are asymptotically tracked, but it does not ensure the non-violation of friction constraints by the contact forces. To guarantee that the contact force vectors remain in the friction cone during the manipulation, a force distribution problem must be solved at each time instant. The force closure assumption ensures that a solution exists that satisfies the friction constraints during the manipulation. This solution, which becomes the reference for the internal force control, can be found with an efficient algorithm, based on the minimization of a convex function that checks the force closure property at each time instant (Bicchi 1995).

Summary and Future Directions

The basic foundation of grasp control has been reviewed with a particular attention to modeling of contact constraints, force closure, and control of object motion and internal forces. This entry did not explicitly address grasp stability that is often equated to grasp closure, because all external forces can be balanced by the hand. A more formal analysis of grasp stability in terms of deflection from an equilibrium point has been proposed for hands with general kinematics in Jen et al. (1996).

The computed-torque control is a classical approach to the grasp control. For a deeper study of other approaches to grasp control based on passivity theory, the reader is referred to Wimboeck et al. (2011).

Recent developments in underactuated robotic hands Birglen et al. (2008) have led to a renewed interest in grasp control. Designing hand with a lower number of actuators has a lot of advantages in terms of robustness and reliability but dramatically reduces the dexterous manipulation abilities which can be recovered only by designing new control algorithms (Prattichizzo et al. 2013).

Cross-References

▸ Force Control in Robotics
▸ Parallel Robots

Recommended Reading

Grasp synthesis and dexterous manipulation are important research topics. Grasp synthesis is the problem of choosing the posture of the hand and contact point locations to optimize a grasp quality metric. One of the first studies of grasp synthesis for multi-fingered hands was undertaken in Jameson (1985) where the author proposed a Levenberg-Marquardt algorithm to search the surface of an object for the locations of three points that would achieve force closure. Since this work, many other metrics and approaches to searching for high-quality grasps have been implemented as discussed in Nguyen (1988), Pollard (1997), Park and Starr (1992), Chen and Burdick (1993), and references therein.

Dexterous manipulation is the capability of manipulating an object so as to arbitrarily steer its configuration in space. Research on dexterous manipulation first appeared in Hanafusa and Asada (1979) where the authors developed a plan to turn a nut onto a bolt. Since then a progression of increasingly complex manipulation tasks have been studied to varying degrees of detail. For the planar case the reader is referred to Mason (1982), Brost (1991), Peshkin and Sanderson (1988), Lynch (1996), and references therein. Several approaches have been proposed to planning and execute dexterous manipulation tasks in three dimensions continues in Cherif and Gupta (1999), Han et al. (2000), and Higashimori et al. (2007). Dexterous manipulation can be evaluated with manipulability ellipsoids of velocity and force as proposed in Chiacchio et al. (1991) for multiple-fingered systems and more recently in Prattichizzo et al. (2012) for underactuated robotic hands.

Bibliography

Bicchi A (1995) On the closure properties of robotic grasping. Int J Robot Res 14(4):319–334

Bicchi A, Kumar V (2000) Robotic grasping and contact: a review. In: Proceedings of the IEEE international conference on robotics and automation, San Francisco, pp 348–353

Bicchi A, Prattichizzo D (1998) Manipulability of co-operating robots with passive joints. In: Proceedings of the IEEE international conference on robtotics and automation, Leuven, pp 1038–1044

Birglen L, Gosselin CM, Laliberté T (2008) Underactuated robotic hands, vol 40. Springer, Berlin

Brost RC (1991) Analysis and planning of planar manipulation tasks. Carnegie Mellon University Pittsburgh, PA, USA. PhD thesis

Brost RC, Goldberg KY (1996) A complete algorithm for designing planar fixtures using modular components. IEEE Trans Robot Autom 12(1):31–46

Chen IM, Burdick JW (1993) Finding antipodal point grasps on irregularly shaped objects. IEEE Trans Robot Autom 9(4):507–512

Cherif M, Gupta KK (1999) Planning quasi-static fingertip manipulation for reconfiguring objects. IEEE Trans Robot Autom 15(5):837–848

Chiacchio P, Chiaverini S, Sciavicco L, Siciliano B (1991) Global task space manipulability ellipsoids for multiple-arm systems. IEEE Trans Robot Autom 7(5):678–685

Ferrari C, Canny J (1992) Planning optimal grasps. In: Proceedings of the IEEE international conference on robtotics and automation, Nice. IEEE, pp 2290–2295

Goldfeder C, Ciocarlie M, Peretzman J, Hao Dang, Allen PK (2009) Data-driven grasping with partial sensor data. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS 2009), St. Louis, pp 1278–1283

Han L, Li Z, Trinkle JC, Qin Z, Jiang S (2000) The planning and control of robot dexterous manipulation. In: Proceedings of the IEEE international conference on robotics and automation, San Francisco, pp 263–269

Han L, Trinkle JC (1998) Dextrous manipulation by rolling and finger gaiting. In: Proceedings of the IEEE international conference on robtotics and automation, Leuven, vol 1. IEEE, pp 730–735

Hanafusa H, Asada H (1979) Handling of constrained objects by active elastic fingers and its applications to assembly. Trans Soc Instrum Control Eng 15(1):61–66

Higashimori M, Kimura M, Ishii I, Kaneko M (2007) Friction independent dynamic capturing strategy for a 2D stick-shaped object. In: Proceedings of the IEEE international conference on robotics and automation, Roma, pp 217–224

Jameson J (1985) Analytic techniques for automated grasp. Department of Mechanical Engineering, Stanford University. PhD thesis

Jen F, Shoham M, Longman RW (1996) Liapunov stability of force-controlled grasps with a multi-fingered hand. Int J Robot Res 15(2):137–154

Johansson RS, Edin BB (1991) Mechanisms for grasp control. In: Pedotti A, Ferrarin M (eds) Restoration of walking for paraplegics. Recent advancements and

R

trends, 3rd edn. Edizioni Pro Juventute/IOS, Milano,
pp 57–65

Li Z, Hsu P, Sastry S (1989) Grasping and coordinated
manipulation by a multifingered robot hand. Int J
Robot Res 8(4):33–50

Lynch K (1996) Nonprehensile manipulation: mechanics
and planning. Carnegie Mellon University School of
Computer Science, March. PhD thesis

Mason MT (1982) Manipulator grasping and pushing op-
erations. PhD thesis, Massachusetts Institute of Tech-
nology, June 1982. Reprinted in Robot hands and the
mechanics of manipulation. MIT, Cambridge

Mason MT, Salisbury JK (1985) Robot hands and the
mechanics of manipulation. MIT, Cambridge

Montana DJ (1988) The kinematics of contact and grasp.
Int J Robot Res 7(3):17–32

Murray RM, Li Z, Sastry SS (1994) A mathematical intro-
duction to robotic manipulation. CRC, Boca Raton

Namiki A, Imai Y, Ishikawa M, Kaneko M (2003) De-
velopment of a high-speed multifingered hand system
and its application to catching. In: Proceedings of
the IEEE/RSJ international conference on intelligent
robots and systems, Las Vegas, vol 3. IEEE, pp 2666–
2671

Nguyen V (1988) Constructing force-closure grasps. Int J
Robot Res 7(3):3–16

Okamura AM, Smaby N, Cutkosky MR (2000) An
overview of dexterous manipulation. In: Proceedings
of the IEEE international conference on robotics and
automation, San Francisco, pp 255–262

Park YC, Starr GP (1992) Grasp synthesis of polygonal
objects using a three-fingered robot hand. Int J Robot
Res 11(3):163–184

Peshkin MA, Sanderson AC (1988) Planning robotic ma-
nipulation strategies for workpieces that slide. IEEE J
Robot Autom 4(5):524–531

Pollard NS (1997) Parallel algorithms for synthesis of
whole-hand grasps. In: Proceedings of the IEEE in-
ternational conference on robotics and automation,
Albuquerque

Prattichizzo D, Bicchi A (1997) Consistent task specifica-
tion for manipulation systems with general kinematics.
J Dyn Syst Meas Control 119(4):760–767

Prattichizzo D, Malvezzi M, Gabiccini M, Bicchi A
(2012) On the manipulability ellipsoids of underactu-
ated robotic hands with compliance. Robot Auton Syst
60(3):337–346. Elsevier

Prattichizzo D, Malvezzi M, Gabiccini M, Bicchi A
(2013) On motion and force controllability of pre-
cision grasps with hands actuated by soft synergies.
IEEE Trans Robot 29(6):1440–1456

Prattichizzo D, Trinkle JC (2008) Grasping. In: Siciliano
B, Kathib O (eds) Handbook of robotics. Springer,
Berlin, pp 671–700

Reuleaux F (1876) The kinematics of
machinery. Macmillan, London. Republished by
Dover, New York, 1963

Roa MA, Suarez R (2009) Computation of independent
contact regions for grasping 3-D objects. IEEE Trans
Robot 25(4):839–850

Salisbury JK, Roth B (1983) Kinematic and force analysis
of articulated mechanical hands. J Mech Trans Autom
Des 105(1):35–41

Saxena A, Driemeyer J, Ng AY (2008) Robotic grasp-
ing of novel objects using vision. Int J Robot Res
27(2):157–173

Shimoga KB (1996) Robot Grasp synthesis algorithms: a
survey. Int J Robot Res 15(3):230–266

Wimboeck T, Ott C, Albu-Schaffer A, Hirzinger G
(2011) Comparison of object-level grasp controllers
for dynamic dexterous manipulation. Int J Robot Res
31(1):3–23

Robot Motion Control

Mark W. Spong
Erik Jonsson School of Engineering and
Computer Science, The University of Texas at
Dallas, Richardson, TX, USA

Abstract

The motion control problem for robots, both for
manipulator arms and for wheeled mobile robots,
is to determine a time sequence of control inputs
to achieve a desired motion, or output, response.
The control inputs are usually motor currents
but can be translated into torques or velocities
for the purpose of control design. The desired
motion is typically given by a reference trajec-
tory, consisting of positions and velocities that
are generated from motion planning and trajec-
tory generation algorithms designed to calculate
collision-free paths, taking into account various
kinematic and dynamic constraints on the robot.
In this chapter we give an overview of some
common control methods for motion control of
robots, concentrating on the control of manipula-
tor arms.

Keywords

Adaptive control; Feedback linearization; Inverse
kinematics; Motion planning; Passivity-based
control; PID control; Robust control

Introduction

We consider the motion control problem for an *n*-degree-of-freedom robot manipulator, such as shown schematically in Fig. 1. The variables $\theta_1, \ldots, \theta_n$ are the *joint variables*, which define the *configuration* of the robot at each instant of time.

A robot manipulator is fundamentally a positioning device *designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks (Robot Institute of America, 1980)*. Thus, manipulator tasks, such as materials transfer, welding, and painting, and even tasks involving the control of interaction forces, such as assembly or grinding, are performed through the coordination and control of the motion of the joints of the robot.

A typical robot control architecture is shown in Fig. 2, which is designed to translate *sensing* into *action*, through motion planning, trajectory generation, and feedback control. In this entry we concentrate on the function of the controller.

Motion Planning

The desired joint motions are specified as reference trajectories (positions and velocities) generated from motion planning algorithms that must determine collision-free paths taking into account various kinematic and dynamic constraints on the robot (Lavelle 2006). A detailed discussion of motion planning is outside the scope of this entry. The motion planning problem begins by decomposing a given task into a discrete set of end-effector motions. A continuous path for the end-effector in the *task space* is then computed, taking into account issues of joint limits and collisions with objects in the workspace, including self-collisions.

Finding optimal paths in configuration space is computationally complex, and methods have been developed to determine feasible, suboptimal paths using various methods such as artificial potential functions, grid search, and roadmaps (Lavelle 2006).

Once a feasible path in task space is determined, a *trajectory*, which is a time-parameterized function in task space or configuration space, is computed. To compute configuration space or joint space trajectories from task space trajectories, the *inverse kinematics* of the manipulator is used.

Trajectory Generation

To simplify computation, joint-level trajectories are typically generated by calculating the inverse kinematics only at discrete points along the task space trajectory and then interpolating between these points. Two of the most common interpolation schemes utilize either *polynomials* in time or *trapezoidal velocity* profiles.
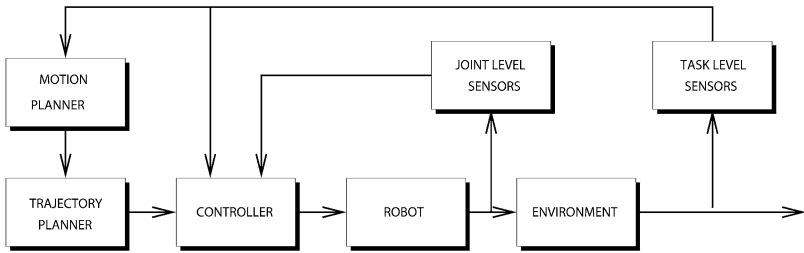
For example, a *cubic polynomial* reference trajectory, $\theta^r(t)$, may be specified as
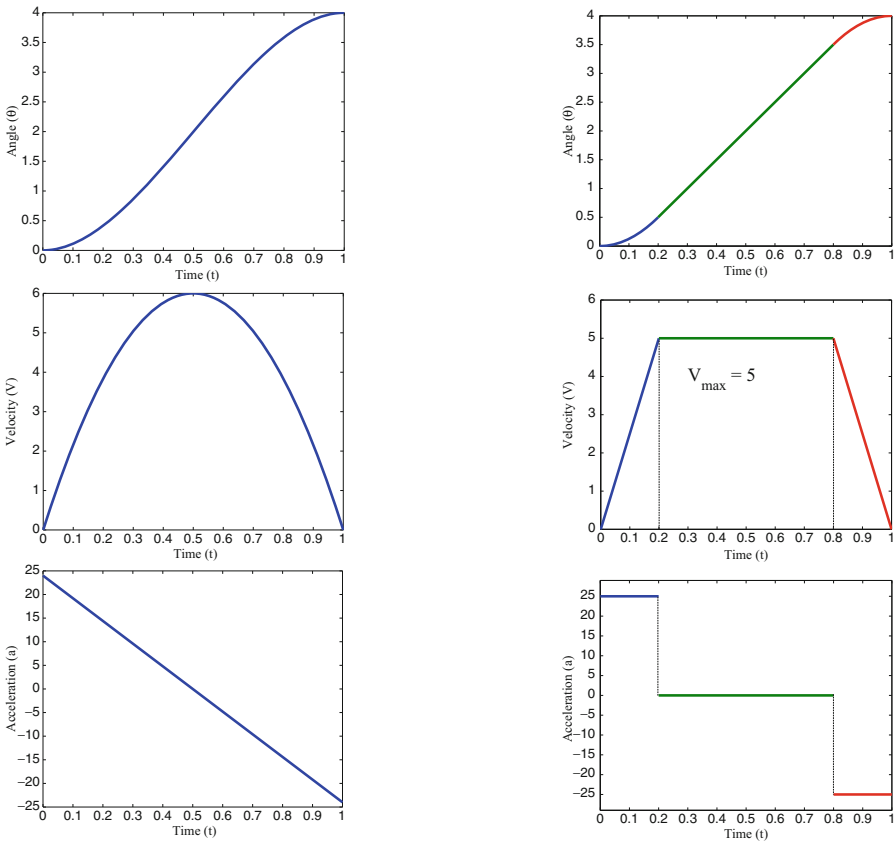
$$\theta^r(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
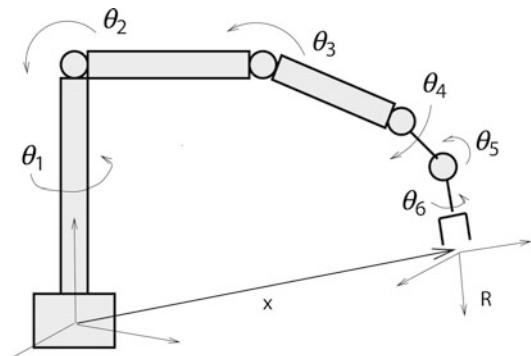
Robot Motion Control,
Fig. 1 Six-link robot
manipulator

Robot Motion Control, Fig. 2 Control architecture for robot control

Robot Motion Control, Fig. 3 A cubic polynomial reference trajectory

Robot Motion Control, Fig. 4 Trapezoidal velocity profile

If the desired positions and velocities of the joint variable are specified at initial and final times, t_0 and t_f, respectively, it is a simple calculation to determine the four polynomial coefficients, $a_0, \ldots, a_3$. The reference velocity and acceleration are then given by

$$\dot{\theta}^r(t) = a_1 + 2a_2 t + 3a_3 t^2$$

$$\ddot{\theta}^r(t) = 2a_2 + 6a_3 t$$

A typical cubic polynomial trajectory is shown in Fig. 3.

A trapezoidal velocity profile is illustrated in Fig. 4.

In this case, the velocity of the joint angle increases linearly to a maximum value, $V_{\max}$,

which remains constant for a period of time and then decreases linearly.

Independent Joint Control

The simplest approach to control design for a multi-degree-of-freedom manipulator is to treat each link of the robot as a single-input/single-output (SISO) system and design the controllers independently for each link. *Proportional, integral, derivative (PID) control* is the most common method employed in this case. This approach works well for highly geared manipulators moving at relatively low speeds, since the large gear reduction and low speed tend to reduce the coupling effects among the various links. More advanced linear or nonlinear control methods can be used to achieve higher performance at the expense of added complexity of the control system.

The basic architecture of such a system, using a linear model to represent the dynamics of each joint of the robot, is shown in the frequency domain in Fig. 6.

The control design objective is to choose the compensator in such a way that the plant output θ *tracks* or follows a desired output, given by the reference signal, θ^r. The control signal, however, is not the only input acting on the system. *Disturbances*, which are really inputs that we do not control, also influence the behavior of the output. Therefore, the controller must be designed, in addition, so that the effects of the disturbance, D, on the plant output are reduced. If this is accomplished, the plant is said to *reject* the disturbances. The twin objectives of *tracking* and *disturbance rejection* are central to any control methodology.

The *plant* transfer function, $P(s)$, represents the dynamics of a single degree-of-freedom system, typically inertia and damping,

$$P(s) = \frac{1}{Js^2 + Bs} \quad (1)$$

$C(s)$ is a PID compensator

$$u(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) (\theta^r(s) - \theta(s)) \quad (2)$$

where K_p, K_i, K_d are the proportional, integral, and derivative gains, respectively, and $\theta^r(s) - \theta(s)$ is the tracking error between the reference trajectory $\theta^r(s)$ and joint variable $\theta(s)$.

Set-Point Tracking

If the reference trajectory θ^r is a constant set point, then the closed-loop transfer function, $T(s)$, from θ^r to θ (with $D = 0$) is

$$T(s) = \frac{P(s)C(s)}{1 + P(s)C(s)}$$

$$= \frac{K_d s^2 + K_p s + K_i}{Js^3 + (B + K_d)s^2 + K_p s + K_i}$$

Applying the Routh-Hurwitz criterion, it follows that the closed-loop system is stable if the gains are positive and

$$K_i < \frac{(B + K_d)K_p}{J} \quad (3)$$

In addition, the presence of the integral control term, $\frac{K_i}{s}$, guarantees zero steady-state error to a constant disturbance term D.

Feedforward Control

In order to track nonconstant reference signals, such as a cubic polynomial trajectory or trapezoidal velocity trajectory, a feedforward term may be superimposed on the PID control signal as shown in Fig. 5. Under the condition that the plant $P(s)$ is *minimum phase*, the feedforward transfer function $F(s)$ can be taken as $1/P(s)$, the inverse of the plant. This guarantees asymptotic tracking of any time-varying reference trajectory provided the closed-loop transfer function is stable.

PID control is, by far, the most common type of control used in industry due to its simplicity. The main problem in implementing PID control is in the *tuning*, that is, in the choice of the
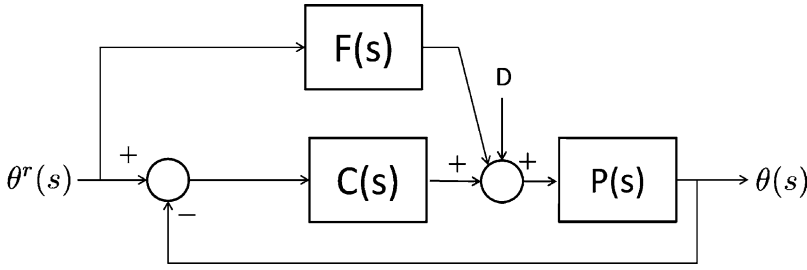
Robot Motion Control, Fig. 5 Feedforward control architecture
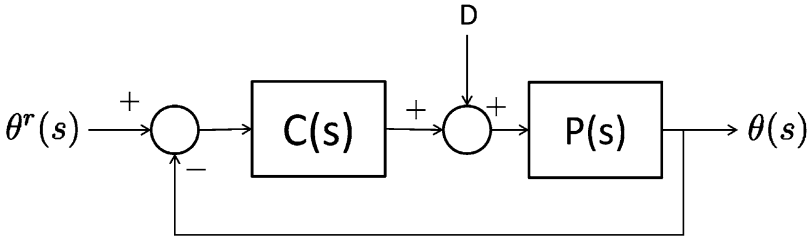
Robot Motion Control, Fig. 6 Single-axis control

proportional, derivative, and integral gains. As we see from the inequality (3), the magnitude of the integral gain K_i is limited by the stability constraint. Therefore, one common design rule of thumb is to first set $K_i = 0$ and design the proportional and derivative gains, K_p and K_d, to achieve the desired transient behavior (*rise time*, *settling time*, and so forth) and then to choose K_i within the limits imposed by (3) to remove the steady-state error.

Advanced Control Methods

Advanced control methods for robots generally aim to take into account issues such as dynamic coupling between joints; compliance in the joints or links; uncertainty in the inertia parameters, such as the masses and moments of inertia of the links; and robustness to sensor noise and other effects. A common model of the dynamics of n-link, rigid robots, i.e., without consideration of friction, elasticity in the joints or links, and other effects, is given by the so-called *Euler-Lagrange* equations

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = \tau \qquad (4)$$

where $\theta = (\theta_1, \theta_2, \ldots, \theta_n)^T$ is the vector of configuration (joint) variables as in Fig. 1. The n-dimensional vectors, $\dot{\theta}$ and $\ddot{\theta}$, are then the joint velocities and accelerations, respectively. The $n \times n$ matrix, $M(\theta)$, is called the inertia matrix. The vectors $C(\theta, \dot{\theta})\dot{\theta}$ and $g(\theta)$ are Coriolis and centrifugal forces and gravitational forces, respectively.

Equation (4) is a system of n coupled, nonlinear, second-order equations and is, in fact, a representation of Newton's Second Law of Motion, where the (generalized) forces acting on the joints of the robot $(\tau - C(\theta, \dot{\theta})\dot{\theta} - g(\theta))$ equate to the mass times acceleration, given by $M(\theta)\ddot{\theta}$.

In this case, the control problem becomes one of choosing the control input torque vector $\tau(t)$, as a function of time, so that the solution, $(\theta(t), \dot{\theta}(t))$, of Eq. (4) tracks a reference trajectory of joint positions and velocities, $(\theta^r(t), \dot{\theta}^r(t))$.

Feedback Linearization Control

An intuitive method of control for this system is the method of *feedback linearization*, which computes the input torque τ according to

$$\tau = M(\theta)a + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) \qquad (5)$$

$$a = \ddot{\theta}^r + K_d(\dot{\theta}^r - \dot{\theta}) + K_p(\theta^r - \theta) \quad (6)$$

with K_d, K_p matrices of appropriate velocity and position error gains.

The control law given by Eqs. (5) and (6) is often referred to as the method of *inverse dynamics* although historically, the method of inverse dynamics control was implemented as a feedforward control

$$\tau = M(\theta^r)a + C(\theta^r, \dot{\theta}^r)\dot{\theta}^r + g(\theta^r) \quad (7)$$

$$a = \ddot{\theta}^r + K_d(\dot{\theta}^r - \dot{\theta}) + K_p(\theta^r - \theta) \quad (8)$$

using the reference position and velocity in place of the measured state. The primary reason for implementing the inverse dynamics in this fashion was the lack of sufficiently fast computation to enable computation of the terms in Eq. (5) in real time. The nonlinearities in Eq. (7) could be precomputed offline and stored to facilitate real-time implementation. With the advent of faster computers, the feedback linearization control is now feasible in real time.

Equations (5) and (6) form a so-called *inner-loop/outer-loop* architecture (Fig. 7). The significance of this architecture is that the nonlinear inner-loop control term (5) results in a linear system with input a and output θ. The design of the outer-loop control can then take advantage of control methods for linear systems. In fact, the control (6) in this case is simply a PD control with feedforward acceleration.

The result of the controller (5) and (6) is a closed-loop system in terms of the tracking error, $e(t) = \theta(t) - \theta^r(t)$, that satisfies the linear equation

$$\ddot{e} + K_d\dot{e} + K_p e = 0 \qquad (9)$$

and therefore, the tracking error converges exponentially to zero for any given reference trajectory.

Task Space Linearization

The inner-loop/outer-loop control architecture above can be modified to track trajectories directly in the task space. Moreover, one can achieve task space tracking by modifying only the outer-loop control a in Eq. (6) while leaving the inner-loop control (5) unchanged. Let $X \in R^6$ represent the end-effector position and orientation and let $X^r(t)$ be a reference trajectory in task space. Since X is a function of the joint variables θ, we have

$$\dot{X} = J(\theta)\dot{\theta} \qquad (10)$$

$$\ddot{X} = J(\theta)\ddot{\theta} + \dot{J}(\theta)\dot{\theta} \qquad (11)$$

where J is the manipulator *Jacobian*. If we now choose the outer-loop term a according to

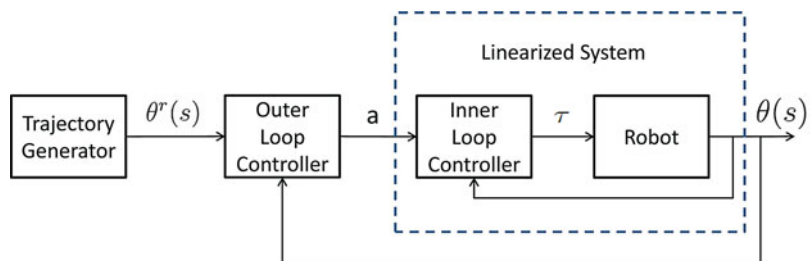$$a = J^{-1}\{a_X - \dot{J}\dot{\theta}\} \qquad (12)$$

with

$$a_X = \ddot{X}^r - K_0(X - X^r) - K_1(\dot{X} - \dot{X}^r) \quad (13)$$

we see that the result is a linear system in the task space tracking error $\widetilde{X}(t) = X(t) - X^r(t)$

$$\ddot{\widetilde{X}} + K_1\dot{\widetilde{X}} + K_0\widetilde{X} = 0 \qquad (14)$$

Robot Motion Control,
Fig. 7 Inner-loop/
outer-loop control
architecture

Therefore, a modification of the outer-loop control achieves a linear and decoupled system directly in the task space coordinates without the need to compute a joint space trajectory and without the need to modify the nonlinear inner-loop control.

It is important to note that the above result is valid in the case of six degree-of-freedom manipulators when the Jacobian J is square and invertible. The case when the Jacobian is not invertible, for example, at kinematic singularities, or when the number of joints is not equal to the dimension of the task space is outside the scope of this entry.

Robust and Adaptive Control

There are several theoretical and practical challenges to the method of feedback linearization control discussed in the previous section. For example, in order to compute Eq. (5), one must have exact knowledge of the parameters defining Eq. (4). In addition, effects of compliance, friction, and so on are not modeled by Eq. (4) and so the stability and performance of the system predicted by Eq. (9) may not be achieved in practice. This has stimulated a great deal of research into robust and adaptive control, control of elasticity, and other issues.

In distinguishing between robust control and adaptive control, we follow the commonly accepted notion that a robust controller is a fixed controller designed to satisfy performance specifications over a given range of uncertainties, whereas an adaptive controller incorporates some sort of online parameter estimation. This distinction is important. For example, in a repetitive motion task, the tracking errors produced by a fixed robust controller would tend to be repetitive as well, whereas tracking errors produced by an adaptive controller might be expected to decrease over time as the plant and/or control parameters are updated based on runtime information. At the same time, adaptive controllers that perform well in the face of parametric uncertainty may not perform well in the face of other types of uncertainty such as external disturbances or unmodeled dynamics.

Robust Feedback Linearization

If we denote by $\hat{M}(\theta)$, $\hat{C}(\theta, \dot{\theta})$, and $\hat{g}(\theta)$ expressions for the terms $M(\theta)$, $C(\theta, \dot{\theta})$, and $g(\theta)$ in the equations of motion (4) based on nominal or estimated values of the true parameters, we can define a control input

$$u = \hat{M}(\theta)(a + \delta a) + \hat{C}(\theta, \dot{\theta})\dot{\theta} + \hat{g}(\theta) \quad (15)$$

where a is as defined in Eq. (6) and δa represents an additional control intended to compensate for the parameter uncertainty. This leads to the state space model in terms of the tracking error e

$$\dot{e} = Ae + B\{\delta a + \eta\}$$

where η represents the *uncertainty* resulting from inexact cancellation of nonlinearities and

$$A = \begin{bmatrix} 0 & I \\ -K_0 & -K_1 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

Under the assumption that the uncertainty is bounded as $||\eta|| \leq \rho(e, t)$, the control term δa can be chosen as

$$\delta a = \begin{cases} -\rho(e, t)\dfrac{B^T Pe}{||B^T Pe||} & ; \text{ if } ||B^T Pe|| > \epsilon \\[2em] -\dfrac{\rho(e, t)}{\epsilon} B^T Pe & ; \text{ if } ||B^T Pe|| \leq \epsilon \end{cases}$$

The Lyapunov function

$$V = e^T Pe \quad (16)$$

where P is a symmetric, positive definite matrix satisfying a Lyapunov equation

$$A^T P + PA + Q = 0 \quad (17)$$

for a given symmetric positive definite matrix Q can be used to show uniform ultimate boundedness of all trajectories, where the size of the

ultimate boundedness set depends on ϵ. This is a practical notion of asymptotic stability in the sense that the tracking errors can be made small.

Passivity-Based Control

Passivity-based control is an alternative to feedback linearization control considered previously and relies on some fundamental structural properties of the Euler-Lagrange equations, primarily *linearity in the parameters* and *passivity*.

The *passivity property* (Ortega and Spong 1989) of robot dynamics follows from the fact that the matrix $N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric, that is, the components n_{jk} of N satisfy $n_{jk} = -n_{kj}$ (Spong et al. 2006). This property implies that the total energy E of the robot satisfies

$$\dot{E} = \dot{\theta}^T u \qquad (18)$$

and can be used to design provably correct robust and adaptive control laws.

Linearity in the Parameters

The robot equations of motion are defined in terms of certain parameters, such as link masses, moments of inertia, etc. The complexity of the dynamic equations makes the determination of these parameters a difficult task. Fortunately, the equations of motion are linear in these inertia parameters in the following sense: There exists an $n \times \ell$ matrix function $Y(q, \dot{q}, \ddot{q})$ and an ℓ-dimensional constant vector Φ such that the Euler-Lagrange equations can be written as

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + g(\theta) = Y(\theta, \dot{\theta}, \ddot{\theta})\Phi \quad (19)$$

The function $Y(\theta, \dot{\theta}, \ddot{\theta})$ is called the *regressor* and $\Phi \in R^\ell$ is the *parameter vector*. The dimension of the parameter space, that is, the number of parameters needed to write the dynamics in this way, is not unique, and finding a minimal set of parameters that can parameterize the dynamic equations is difficult in general.

Passivity-Based Robust Control

The passivity and linearity-in-the-parameters properties of the robot dynamics can be exploited to design robust and adaptive controllers that do not attempt to cancel the system nonlinearities as in the inverse dynamics approach. A passivity-based robust controller may be defined as

$$u = \hat{M}(\theta)a + \hat{C}(\theta, \dot{\theta})v + \hat{g}(\theta) - Kr \quad (20)$$

where the quantities v, a, and r are given as

$$v = \dot{\theta}^r - \Lambda\tilde{\theta}$$
$$a = \dot{v} = \ddot{\theta}^r - \Lambda\dot{\tilde{\theta}}$$
$$r = \dot{\theta} - v = \dot{\tilde{\theta}} + \Lambda\tilde{\theta}$$

and K is a diagonal matrix of positive gains.

Using the linearity-in-the-parameters property, the closed-loop system can be written as

$$M(\theta)\dot{r} + C(\theta, \dot{\theta})r + Kr = Y(\theta, \dot{\theta}, a, v)$$
$$(\hat{\Phi} - \Phi) \qquad (21)$$

In the robust passivity-based approach, the term $\hat{\Phi}$ is chosen as

$$\hat{\Phi} = \Phi_0 + \delta\Phi$$

where Φ_0 is a fixed nominal parameter vector and $\delta\Phi$ is an additional control term. The additional term $\delta\Phi$ can be designed according to

$$\delta\Phi = \begin{cases} -\rho\dfrac{Y^T r}{||Y^T r||} & ; \text{ if } ||Y^T r|| > \epsilon \\[2ex] -\dfrac{\rho}{\epsilon}Y^T r & ; \text{ if } ||Y^T r|| \leq \epsilon \end{cases}$$

where ρ is a (constant) bound on the parameter uncertainty. Uniform ultimate boundedness of the tracking errors follows using the Lyapunov function

$$V = \frac{1}{2}r^T M(\theta)r + \tilde{\theta}^T \Lambda K \tilde{\theta}$$

where, as before, the size of the ultimate boundedness set depends on the parameter ϵ.

Passivity-Based Adaptive Control

In the adaptive version of this approach, we consider again the control law (20) and the resulting closed-loop system

$$M(\theta)\dot{r} + C(\theta, \dot{\theta})r + Kr = Y(\hat{\Phi} - \Phi)$$

In this case, the term $\hat{\Phi}$ is taken as the output of an estimator

$$\dot{\hat{\Phi}} = -\Gamma^{-1}Y^T(\theta, \dot{\theta}, a, v)r \qquad (22)$$

The Lyapunov function

$$V = \frac{1}{2}r^T M(\theta)r + \tilde{\theta}^T \Lambda K \tilde{\theta} + \frac{1}{2}\tilde{\Phi}^T \Gamma \tilde{\Phi}$$

can be used to show global convergence of the tracking errors to zero and boundedness of the parameter estimates.

One of the problems with the adaptive control approaches considered here is the so-called *parameter drift* problem. It can be shown that the estimated parameters converge to the true parameters provided the reference trajectory satisfies the condition of *persistency of excitation*

$$\alpha I \leq \int_{t_0}^{t_0+T} Y^T(\theta_r, \dot{\theta}_r, \ddot{\theta}_r)Y(\theta^r, \dot{\theta}^r, \ddot{\theta}^r)dt \leq \beta I$$

for all t_0, where α, β, and T are positive constants.

Summary and Future Directions

We have discussed the commonly applied methods of PID control, feedback linearization control, as well as robust and adaptive control for motion control of robot manipulators. There is a large and relatively mature body of literature on these methods, and in fact, the material here is now contained in standard textbooks in robotics, such as Siciliano et al. (2010) and Spong et al. (2006).

Future directions in robot motion control include the full integration of vision, force, and position feedback, cooperative control of multiple arms, and advances in machine learning and human-robot interaction. Direct control of robots through brain-machine interfaces is also an active area of research and will enable new areas of applications such as medical assistive robots.

Cross-Referenes

- ▶ Adaptive Control, Overview
- ▶ Cooperative Manipulators
- ▶ Flexible Robots
- ▶ Force Control in Robotics
- ▶ Lyapunov's Stability Theory
- ▶ Robot Teleoperation

Recommended Reading

Many of the fundamental theoretical problems in motion control of robot manipulators were solved during an intense period of research from about the mid-1980s until the early-1990s during which time researchers first began to exploit the structural properties of manipulator dynamics such as feedback linearizability, skew symmetry and passivity, multiple time-scale behavior, and other properties. For a more advanced treatment of some of these topics, the reader is referred to Spong et al. (1992) and Canudas de Wit et al. (1996).

A survey of robust control of robots up to about 1990 is found in Abdallah et al. (1991). The passivity-based robust control result here is due to Spong (1992). The first results in passivity-based adaptive control of manipulators were in Horowitz and Tomizuka (1986) and Slotine and Li (1987). The Lyapunov stability proof

of passivity-based adaptive control is due to Spong et al. (1990). A unifying treatment of adaptive manipulator control from a passivity perspective was presented in Ortega and Spong (1989).

Bibliography

Abdallah C, Dawson DM, Dorato P, Jamshidi M (1991) A survey of robust control of rigid robots. IEEE Control Syst Mag 11(2):24–30

Canudas de Wit C et al (1996) Theory of robot control. Springer, Berlin

Horowitz R, Tomizuka M (1986) An adaptive control scheme for mechanical manipulators – compensation of nonlinearities and decoupling control. Trans ASME J Dyn Syst Meas Control 108:127–135

Hunt LR, Su R, Meyer G (1983) Design for multi-input nonlinear systems. In: Brockett RW et al (eds) Differential geometric control theory. Birkhauser, Boston/Basel/Stuttgart, pp 268–298

Lavelle SM (2006) Planning algorithms. Cambridge University Press, Cambridge

Markiewicz BR (1973) Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator. NASA-JPL Technical Memo, pp 33–61

Ortega R, Spong MW (1989) Adaptive motion control of rigid robots: a tutorial. Automatica 25(6): 877–888

Siciliano B, Sciavicco L, Villani L, Oriolo G (2010) Robotics: modeling, planning, and control. Springer, London

Slotine J-JE, Li W (1987) On the adaptive control of robot manipulators. Int J Robot Res 6(3): 147–157

Spong MW (1987) Modeling and control of elastic joint robots. Trans ASME J Dyn Syst Meas Control 109:310–319

Spong MW (1992) On the robust control of robot manipulators. IEEE Trans Autom Control AC-37(11):1782–1786

Spong MW, Vidyasagar M (1987) Robust linear compensator design for nonlinear robotic control. IEEE J Robot Autom RA 3(4): 345–350

Spong MW, Ortega R, Kelly R (1990) Comments on 'adaptive manipulator control: a case study'. IEEE Trans Autom Control 35:761–762

Spong MW, Lewis FL, Abdallah CT (1992) Robot control: dynamics, motion planning, and analysis. IEEE, New York

Spong MW, Hutchinson S, Vidyasagar M (2006) Robot modeling and control. Wiley, New York

Tarn TJ, Bejczy AK, Isidori A, Chen Y (1984) Nonlinear feedback in robot arm control. In: Proceedings of the IEEE conference on decision and control, Las Vegas, pp 736–751

Robot Teleoperation

Claudio Melchiorri
Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione, Alma Mater Studiorum Università di Bologna, Bologna, Italy

Abstract

Robots may allow human beings to physically interact with remote objects and environments. This possibility is known as *robot teleoperation* and permits to operate in conditions or environments dangerous for human operators. Although teleoperation was among the first developments in robotics back in the 1950's, still nowadays there are important and difficult challenges for researchers and scientists, showing the intrinsic difficulties of this fascinating field of robotics.

Keywords

Bilateral control; Force reflection; Robot teleoperation; Time delay

Introduction

A robotic teleoperation system allows to reproduce the actions of a human operator and to interact physically with objects and environments placed at a distance. This possibility has always attracted the human being, and telemanipulation has been one of the first fields to be developed in robotics: the first modern applications of this type of technology are dated back to the 1940s and the early 1950s for handling radioactive material (Goertz and Thompson 1954), for underwater and space applications (Martin and Kuban 1985; Vertut and Coiffet 1986), and for human prostheses (Kobrinskii 1960). For an overview on applications, see Sheridan (1992), Hokayem and Spong (2006), Ferre et al. (2007) and the related references. Nevertheless, despite the research interest and the many existing devices, many

challenging problems have still to be fully solved both from the technical and control point of view.

In these notes, an overview on robot teleoperation is presented. In particular, the following points are illustrated:

- General description of a telemanipulation system and of its key components: the "*master*," the "*slave*," and the "*communication channel*"
- Overview on applications and existing devices
- Some control techniques for telemanipulation systems: "traditional" force reflection, shared compliance control, Passivity-based control, predictive control, four-channel architecture

General Description of a Telemanipulation System

A telemanipulator is a complex mechatronic system in which the main elements are a *master* (or local) and a *slave* (or remote) device, interconnected by a *communication channel*. The overall system is interfaced on one side (the master) with a human operator and on the other (the slave) with the environment: see Fig. 1.

Both the master and slave devices have a local controller, with a hardware/software complexity that can be quite different depending on the system and task to be executed. Key features of this type of devices, usually not present in a typical robotic manipulation system, are:

1. A human operator is involved in the loop for the (high-level) control of the task execution.
2. It is necessary to provide to the operator, possibly in real time, data related to the task. This implies the presence of a suitable user interface and the selection of proper signals transmitted to the operator. These signals, e.g., related to forces applied to the environment, relevant positions of the slave, graphical video data, and tactile or acoustic information, have strong implications on the control properties and performances of the system.
3. A communication channel is present between the master and the slave. This channel may represent a source of problems when time-delays are present since, as well known from the control theory, delays in a feedback loop may generate instability. Problems of this type, firstly observed in a force feedback scheme in 1965 (Ferrel 1966), arise, for example, in underwater or space operations. Note that even time-delays of the order of the tenth of a second may create instability problems.
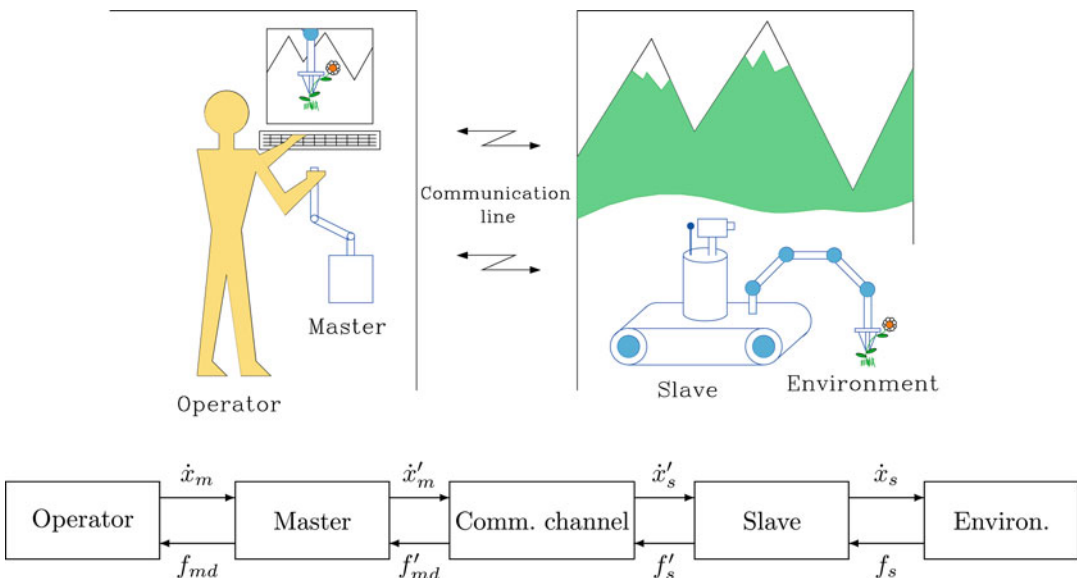
Robot Teleoperation, Fig. 1 A telemanipulation system and its block scheme representation. Subscripts m and s refer to variables at the master and slave site, respectively

In the block diagram of Fig. 1, some implicit choices have been made. The operator specifies a desired velocity ($\dot{x}_m$) to be applied to the environment through the master, the communication channel, and the slave and receives back a force signal (f_{md}). In the figure, the flow of the signals could be reversed as well, letting the operator specify a force to the environment and receiving back a velocity information. This is equivalent to reversing the roles of the master and slave devices. When this operation is possible, the teleoperation system is defined *bilaterally controlled* (Bejczy and Handlykken 1981).

One of the goals of the control system is to have, in steady state, the slave velocity equal to the master velocity, i.e., $\dot{x}_s = \dot{x}_m$, and similarly for the forces, $f_{md} = f_s$. When this is accomplished, the teleoperator is defined *transparent* (Lawrence 1993).

In this general framework, the main features of the components of a telemanipulator are the following.

The Master

The master, or local system, is the interface through which the operator specifies commands to the whole device. Typical features of the master are:

– Capability of assigning tasks to the slave and providing the operator with relevant information about the task development. In fact, an important feature of the master is its capability of providing the operator with the *telepresence*, i.e., the sensation of being in some manner involved with task execution. In this respect, several solutions have been adopted, varying from joysticks and/or consoles (Hirzinger et al. 1992) to exoskeletons (Bergamasco et al. 2007; Smith et al. 1992) and so on. In these devices, different types of signals may be reflected to the operator, from simple graphical data to full kinetostatic information.
– Capability of acquiring and processing data from both the operator and the slave. Typical elaborations are filtering, prediction, delay compensation, modeling of remote and local dynamics, and so on.

The Slave

The slave, or remote system, is the part of the teleoperator which directly interacts with the environment for task execution. Requirements similar to the master may be specified for the slave system:

– A robotic system for the interaction with the environment and the execution of the task planned by the operator. This part, usually provided with autonomous features, has to be in some way customized to operate in particular environments, e.g., submarine, outer space, and nuclear areas. Note that the kinematics and the dynamics of the remote manipulator may be different from those of the local one (when present), originating several problems when telepresence is needed for task execution (Colgate 1993).
– Signal acquisition and processing. Sensory capability is a main requirement for the slave device, which is often equipped with video cameras, force/tactile sensors, proximity sensors, and so on.
– Capability of data processing. Also the remote site must be able to elaborate the information needed for task execution. In fact, besides other considerations, the destabilizing effects caused by communication delays and/or restricted bandwidths of transmission must be compensated locally, providing the slave system with a certain degree of autonomy.

The Communication Line

The communication line represents the link between the master and slave sites. Different platforms may be used for this purpose, from radio connections by means of satellites to cables for underwater operations. The main drawback that can be introduced by this element is a delay, due both to a physical delay in the transmission line (e.g., in a long satellite communication) and to limited bandwidth of the hardware. This delay, that sometimes is not even constant, can cause noticeable instability problems if proper compensating actions are not taken.

An Overview on Applications

Use of telemanipulators, in the broader sense of the terminology, may be found in a number of different applications developed since the early 1950s. First examples of these devices have been designed and realized for operations in radioactive environments and for human limb prostheses. At the moment, this type of technology is applied in a number of different fields: space, underwater, medicine, hazardous environments, security, simulators, and so on.

Space Applications

Robotics is used in space for exploration, scientific experiments, and commercial activities. Main reasons of space telerobotics are the high costs and the hostile environment for human beings. For many years, the main example of teleoperation in space was applications in space shuttle activities where the operators had a direct control of the task executed by the manipulator. Nowadays, an important application of robot technology is for planetary missions, where autonomous telerobots are required and the operator has only a supervisory control of the task. Main directions of current research activity for space robotics are the development of arms for both intra-vehicular and extravehicular activities, free-flying platforms, and planetary rovers.

Among the most known examples of robot arms for space one can list the Canadian Remote Manipulator System (RMS), installed on the US space shuttles. The 6 degree-of-freedom (dof) arm, built by the Canadian firm SPAR, had a flexible, 15 m long structure and was capable of executing preprogrammed and/or teleoperated tasks. Five arms have been built, working on space shuttles from 1981 to 2011. Since 2001 the Canadarm 2 is used on the ISS (International Space Station). This 7 dof, 17.6 m long arm is used for assembly and maintenance purposes.

Concerning planetary exploration, a first successful space telerobotic program has been the Mars Viking Program, which performed scientific experiments on Mars in 1976. More recently, NASA has sent to Mars the rovers Sojourner in 1997 (working for about 3 months) and Spirit and Opportunity, which arrived in 2004. Opportunity is still working (January 2014), see http://marsrovers.jpl.nasa.gov/home/index.html. New missions on Mars with other, more complex, rovers are currently planned by NASA.

With the current technological possibilities, further substantial developments in this field are slowed down by the large amount of money and time required to guarantee a successful mission. However, relevant technical problems still exist due to reliability requirements, weight constraints, hostile environments and communication time-delays (ranging from 1 s in earth orbits to 4–40 min or more for planetary missions).

Underwater

After the first successful military applications of underwater telerobotics (in 1966 the US Navy's CURV – Cable-controlled Underwater Recovery Vehicle – was successfully employed to retrieve a nuclear bomb from the ocean), extensive use of ROVs (remote operated vehicles) has started in the 1980s for offshore operations for oil/gas industry. At the moment, underwater telerobotics is mainly used for business, military missions, and scientific expeditions. Telerobotic (autonomous) tasks are usually limited to small routine tasks rather than complete activities, for example, simple tool switching operations, repetitive bolt/nut screwing, and piloting to new locations. First examples of underwater teleoperation were mainly based on manned submersibles, either free swimming or connected to a surface ship, and with teleoperated arms on the outer structure. In more recent operations, human operators remotely control the submersibles by long fiber-optic cables for data communication, increasing the costs and complexity of the missions.

Probably, the most important users are in the business field, where it is more convenient to use teleoperated devices rather than human divers to perform inspections and repairs on deep sea equipment. The main users of telesubmersibles are the oil and communication (telephone) industries, where underwater pipes and cables require routine operations. The scientific community uses this technology for marine biological,

geological, and archeological missions, while the military have used telerobotics in many salvage operations, such as plane or watercraft recovery.

The conditions of the water environment, e.g., the high pressures, the poor visibility, and the communication difficulties, cause the major problems in this field. In order to solve the problems due to the high pressure, a very robust mechanical structure and (typically) hydraulic actuators are employed. On the other hand, vision problems are not so easily solved, being related to several factors of the environment. External lighting is necessary, and other technologies (e.g., sonar) are sometimes used. Computer graphic simulation may help the user during task execution in partially known environments. For references, see, e.g., Ridao et al. (2007).

Medical Telerobotics

Several teleoperated devices are found in the medical field. In fact, robotic manipulators are used to perform surgery, diagnose illnesses or injuries, help impaired people, and train specialized medical personnel.

Robotic systems of different complexities have been developed since the 1950s for aid to impaired people. Among the most common systems are automated wheelchairs, controlled by voice or by joysticks for hand, mouth, eye, or head movements.

At the moment, there is a relevant interest in applying teleoperated devices in microsurgery operations, e.g., eye surgery, where small precise movements are needed. The movements of the operator are scaled down by the mechanism so that very fine operations can be performed while maintaining a suitable telepresence effect. Another important class of surgical process consists of the so-called minimally invasive procedures. In this case, the surgeon operates through small insertions using thin medical instruments and small video cameras. The increased difficulties for the surgeon are partially compensated by computers, which are used to create virtual environments where the use of telepresence plays a fundamental role.

A very attractive application is the use of telemanipulators in remote surgery operations.

Telediagnosis may also broaden the range of a single doctor by allowing to exam a patient visually or viewing records on a computer interface. Finally, telepresence is becoming very important for the instruction of specialized doctors and to perform rehearsals before the actual operation.

Security

Applications in this area aim to employ telerobotic devices for the protection of persons and properties. Most systems used in this area are teleoperated devices since these tasks require decision capabilities and intelligence levels not currently possible for machines, although the use of autonomous systems is more and more frequent.

In the area of security, robots may be used for patrolling buildings and for protection purposes. These devices can either be autonomous or teleoperated. Military applications adopt principally teleoperation, mainly for locating enemies or dangerous equipment without direct risk for human personnel. Unmanned aeroplanes or teleoperated devices for the detection and destruction of mines or bombs are well-known examples of this technology. Teleoperation is also used for fire extinguishing, in order to spray water or chemical agents with remotely operated vehicles.

Telerobotics in Hazardous Environments

Robots may substitute human beings for operations in hazardous environments; as a matter of fact, nuclear industry was the first important user of modern teleoperating devices. Telerobotics is applied in several nuclear or chemical plants and also for military applications (e.g., for building military equipment and arms) in a variety of tasks. Besides direct handling of radioactive or chemical material, robots are used in waste cleanup/disposal and plant inspection. Ammunition disposal also makes use of telerobotic machines.

Telerobotics in Mining and Other Industries

Besides the typical use of robots in a number of industrial applications (assembly, welding, painting, and so on), other applications of robotic

R

systems in nonconventional production processes have been developed, for example, in mines, constructions, agriculture, warehousing, and many other activities.

Use of telemanipulators for mining applications, despite the relevant motivations such as high costs and risks of human work, finds difficulties and limitations caused by the particular environment and the relevant level of autonomy requested to operate in mines. As a matter of fact, the mining industry has only recently started to experiment teleoperated devices; see e.g., Duff et al. (2010). These machines are being developed to perform frame wall building, structure testing, hole drilling, wall blasting, mine digging, and so on.

In construction tasks, not considering that all construction/destruction machinery controlled by a human can be regarded as examples of teleoperators (e.g., cranes and front-end loaders), applications of real telerobotic systems are not so numerous because of the unstructured environments and the nonrepetitive tasks. Current work in this area concerns the development of machines for earth movement, construction of structures, building window washing, bridge inspection and maintenance, and power line repair.

The Control Problem

For the development of a reliable teleoperation system, providing force feedback to the user, the problems caused by the interaction of the robotic device with the environment and the possible time-delays caused by the communication channel have to be properly considered and solved.

In telemanipulation without either force feedback to the operator or a local compliance control, the remote manipulator is strictly controlled according to the master position signal. As a consequence, the system results in being stiff, and errors between the master and slave positions may cause excessive and undesired contact forces.

In bilateral telemanipulation, it has been proven that a profitable manner for increasing system performances (e.g., in terms of task completion time, total contact time, and cumulative contact force) is to reflect back to the operator information about the force applied to the environment. On the other hand, it results that the force reflection gain, that gives to the operator the feeling of the interaction, destabilizes the system, especially when time-delays are present.

Control schemes for robotic teleoperation devices can be classified according to the general structures reported in Fig. 2, showing the *direct teleoperation*, the *coordinated teleoperation*, and the *supervisory control* schemes. In the direct teleoperation scheme, possible only for negligible time delays, the operator has direct control of the slave robot and receives feedback in real time. In the coordinated teleoperation scheme, the operator still controls the remote robot, but low-level control loops in the slave system are present because time delays do not allow the operator to control directly the actuators. In the supervisory control scheme, the remote site has more autonomy and task execution is controlled locally, while the operator gives mainly high-level commands and acts as a supervisor. A local loop is present also at the master side, indicating the presence of (usually) a model (graphical, mathematical, etc.) of the slave site to improve performances in case of large time delays.

Some of the main control architectures for teleoperation devices presented in literature to deal with the problems of time-delay and force reflection are now briefly described and commented. The considered architectures are the *"traditional" force reflection*, the *shared compliance control*, the *passivity-based teleoperation*, the *predictive control*, and the *four-channel scheme*. However, many other control schemes have been presented in the literature; see, e.g., Arcara and Melchiorri (2002), Hirche et al. (2007), and therefore what is presented here is a brief, though significant, overview in order
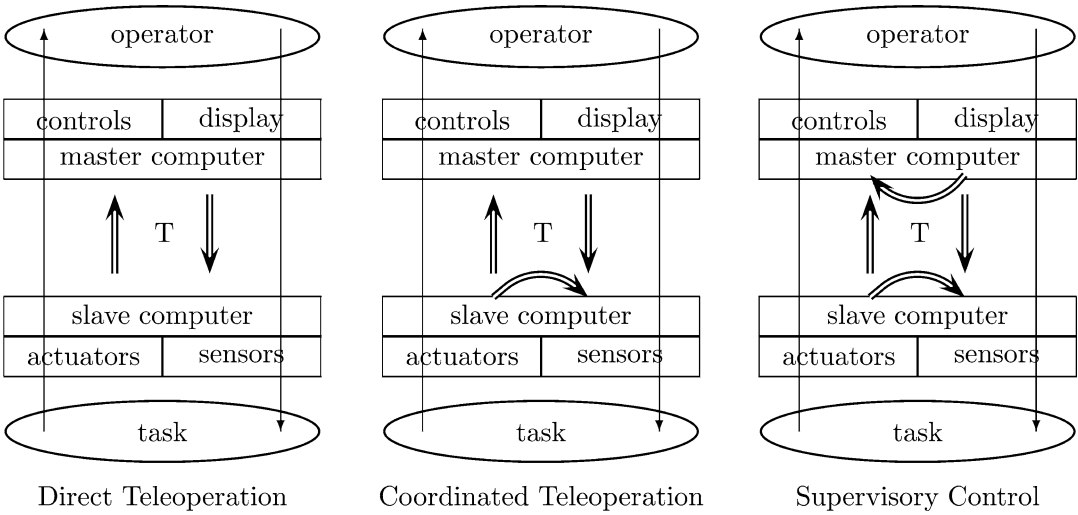
Robot Teleoperation, Fig. 2 Possible structures of bilateral control schemes for robotic teleoperation

Robot Teleoperation, Fig. 3 The "traditional force reflection" transmission scheme
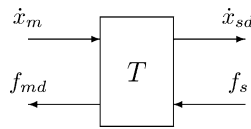
to focus on the major problems encountered in this field and on some of the approaches for their solutions.

Traditional Force Reflection Teleoperation

The simplest manner of transmitting the remote force to the operator is to reflect it directly, without any particular elaboration, as shown in Fig. 3. The resulting transmission equations are

$$\begin{cases} f_{md}(t) = f_s(t - T) \\ \dot{x}_{sd}(t) = \dot{x}_m(t - T) \end{cases} \tag{1}$$

where T is the time-delay introduced by the communication network and subscript d indicates the desired set point for the master (m) and slave (s) controllers.

This technique presents relevant instability problems due to time-delays. As a matter of fact, it is possible to verify that the communication channel does not present strictly passive properties, even for limited bandwidths of the input signals $\dot{x}_m$ and f_s. This result is valid also considering an attenuation between f_{md} and f_s, i.e., introducing a force reflection gain $G_{fr} < 1.0$ in (1) and computing therefore $f_{md}(t) = G_{fr} f_s(t - T)$. The attenuation reduces the telepresence sensation and degrades the performances, but still does not cause a passive (then stable) network. The nonpassive channel has the global effect of introducing in the overall system energy flows that, if not properly reduced by the local controllers, contribute to destabilize the telemanipulator.

The dynamics of the overall system may be described by the following two sets of equations: the first taking into account the master dynamics and the force transmitted by the communication channel and the second including the slave dynamics, the position signals of the channel, and the local position controller:

R

$$\begin{cases} M_m \dot{x}_m(t) = -f_{md}(t) - B_m \dot{x}_m(t) - K_h x_m(t) \\ f_{md}(t) = f_s(t - T) \end{cases} \qquad \begin{cases} M_s \dot{x}_s(t) = f_s(t) - B_s \dot{x}_s(t) \\ \dot{x}_{sd}(t) = S_p \dot{x}_m(t - T) \\ f_s(t) = K_p[x_{sd}(t) - x_s(t)] \end{cases}$$

In the above equations, M_i and B_i, $i = m, s$, are masses and damping factors at the master and slave sites, K_h represents the operator (simply modeled as a stiffness) and K_p the slave position controller. The gain S_p has been added, with respect to Eq. (1), in order to scale velocity variables between the two robotic systems.

It can be shown that this control scheme does not guarantee stability in the presence of time delays, although in practical applications stability may still be achieved for small time delays due to dissipation introduced by friction and the local controllers.

Shared Compliance Control

As previously mentioned, both the interactions of the robotic device with the environment and the effects of time-delays have to be considered in the definition of control strategies for telemanipulation systems. The *position-error based force reflection* scheme deals with both these effects (Kim et al. 1992). This scheme is based on the computation of the feedback signal f_{md} as a force proportional to the error between master and slave positions:

$$f_{md}(t) = G_{fr}[x_m(t) - x_s(t - T)]$$

This signal gives to the operator a sensation related to the difference between the postures of the robotic devices caused either by interactions or delays. Note that in this manner an elastic (proportional) element is introduced between the positions of the robots. This allows to obtain a stable behavior of the overall system comprehensive of local controllers, at least for limited values of G_{fr}.

An additional feature for dealing with problems due to time-delays is the so-called *shared compliance control* (SCC). A local, autonomous force feedback is realized at the slave site in order

to program active compliance and damping of the robotic device. This control action is important when compliance has to be realized between the (stiff) mechanical device and its environment and during the collision or contact phases. The overall control system is therefore based on sharing autonomous and human-driven control actions. A block diagram of the whole system, including the master and slave dynamics ($1/(M_m s^2 + B_m s + K_h)$ and $1/(M_s s^2 + B_s s)$ respectively), the force reflection gain (G_{fr}), the shared compliance controller (G_{cc}), and an environment model (K_e), is shown in Fig. 4.

For a given time-delay, the force reflection gain G_{fr} can be increased with respect to the traditional force reflection scheme. In any case, when the time-delay increases, the gain has to be correspondingly decreased to guarantee stability, i.e., the value of G_{fr} depends directly on the amount of time-delay. In fact, also this control scheme in general does not present passivity features, although it can be shown that it may be stable (for a limited range of time delays) with a proper choice of the control parameters.

Passivity-Based Teleoperation

A control scheme inspired by the passivity theory (Van der Schaft 2000) is now described. Basic consideration is that the communication channel may represent, if proper actions are not taken, a non-passive element between the master and slave. With proper modifications, the transmission line presents passive properties, and therefore, the stability of the overall system may be achieved for any value of the time-delay T.

Lossless Transmission Line

Results of passivity and scattering theories can be used to show that in traditional force reflection teleoperation, Eq. (1), the instability of the overall system in presence of time-delays is caused by
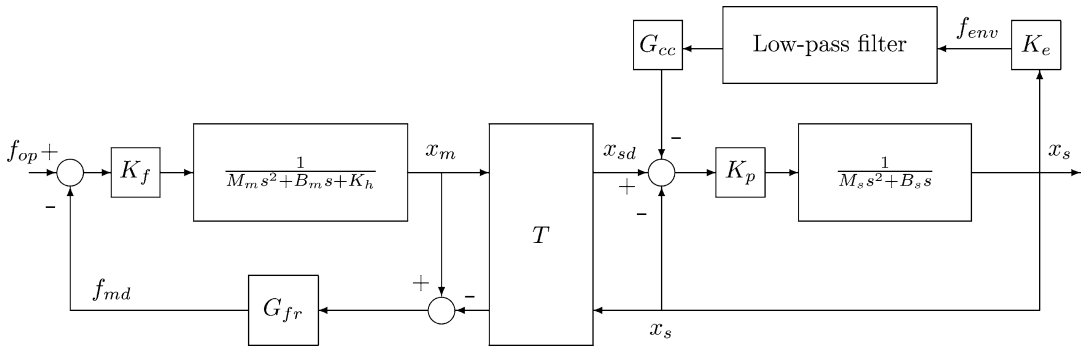
Robot Teleoperation, Fig. 4 Position-error based force reflection with SCC at the remote site

the non passive properties of the communication channel (Niemeyer and Slotine 1991). On the other hand, it has been shown that the definition of a communication network based on a lossless transmission line provides the system with passivity features for any time-delay (Anderson and Spong 1989), facilitating therefore the stability of the overall system.

For the definition of a lossless transmission line, it is convenient to refer, instead of the velocity and force variables $\dot{x}$, f at each port (see Fig. 3), to the equivalent *wave* variables u and v that are related to the passivity formalism and whose definition derives from the theory of electric circuits. By using these variables, it is possible to describe the power balance in a circuit as the difference of two positive terms which consider the input and the output power. In fact, by introducing the input wave $u = [u_m^T, u_s^T]^T$ and the output wave $v = [v_m^T, v_s^T]^T$, the power balance in the teleoperator can be expressed as

$$P = \frac{1}{2}\left(u^T u - v^T v\right) = f^T \dot{x} = [f_m^T, f_s^T]\begin{bmatrix} \dot{x}_m \\ -\dot{x}_s \end{bmatrix}$$

By considering a proper scaling factor b, defined as the *characteristic impedance* of the transmission line, the previous equation defines the following transformations between power and wave variables:

$$u_m = \frac{1}{\sqrt{2b}}(f_m + b\,\dot{x}_m) \quad u_s = \frac{1}{\sqrt{2b}}(f_s - b\,\dot{x}_s)$$
$$v_m = \frac{1}{\sqrt{2b}}(f_m - b\,\dot{x}_m) \quad v_s = \frac{1}{\sqrt{2b}}(f_s + b\,\dot{x}_s)$$

The resulting network is described by

$$\begin{cases} f_{md}(t) = f_s(t - T) + b[\dot{x}_m(t) - \dot{x}_{sd}(t - T)] \\ \dot{x}_{sd}(t) = \dot{x}_m(t - T) + \frac{1}{b}[f_{md}(t - T) - f_s(t)] \end{cases}$$

In terms of wave variables, the passivity-based communication network is described as (see Fig. 5)

$$\begin{cases} f_{md}(t) = b\,\dot{x}_m(t) + \sqrt{2b}\,v_m(t) \\ \dot{x}_{sd}(t) = -\frac{1}{b}[f_s(t) - \sqrt{2b}\,v_s(t)] \end{cases}$$

In analogy with electric networks, impedance adaptation should be added to both extremities of the transmission line, as described e.g., in Niemeyer and Slotine (1991).

Predictive Control

In a well-known example of space telerobotics, the ROTEX project (Hirzinger et al. 1992), the problems introduced by force feedback and time-delays have been solved in a different manner. In fact, in this case the force information is not transmitted to the operator, and an extensive use of graphic simulation and *telesensor programming* is made to help control of the task execution.

In particular, the *predictive display* technique (Sheridan 1992) has been employed for generating and extrapolating beforehand visual indications, such as cursors or wire frame models of the manipulator and its environment. These information are generated by the control system and assist the operator in driving the task
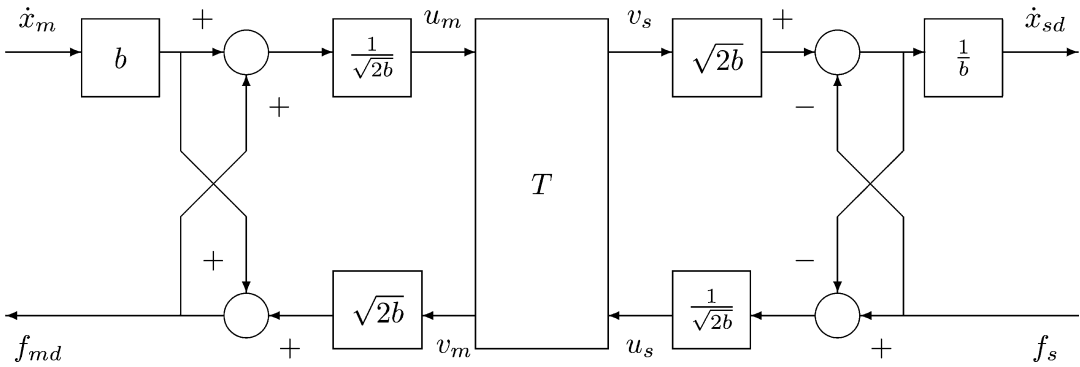
Robot Teleoperation, Fig. 5 Transmission line based on passivity

execution more efficiently. In this case, a proper prediction algorithm has to be set on the basis of current initial conditions of the manipulator and, possibly, of current control variables.

In telerobotics, predictive displays have to be purposely designed in order to consider the prediction of motions of the manipulator. Usually, the task is graphically simulated in real time, without time-delay, exploiting a model of the remote environment and of the slave device. The operator can observe the task executed by the remote system on the screen, where a simulated copy (with $T = 0$ s) of the robotic device can be superimposed on the real operating device in the scene of the remote site. In this manner, the operator may program appropriate actions for the interaction with the environment.

This type of task planning helps when a noticeable time-delay occurs. In fact, when operators deal with relevant time-delays (e.g., larger than 1 s), usually they operate with a "move and wait" strategy, conservatively specifying only small displacements to the remote robot. By using predictive display, the time required to execute complex tasks is greatly reduced. On the other hand, the operator has only visual information about the remote environment and the task execution.
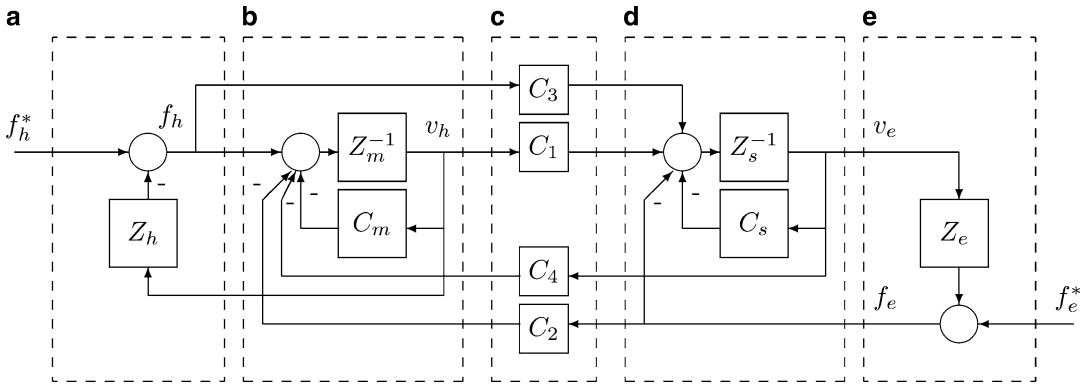
Four-Channel Scheme

A generalization of the scheme of Fig. 1, the so-called four-channel architecture (Hirche et al. 2007; Lawrence 1993), is shown in Fig. 6. In this scheme, both the velocity and force signal of

the master and slave are transmitted, and with a proper choice of the four blocks C_1, C_2, C_3, and C_4 many design goals can be achieved, in particular concerning the stability and the transparency of the overall system. In particular, if $C_3 = C_4 = 0$, the standard velocity-force transmission scheme is obtained, while ideal transparency is achieved if $C_1 = Z_{cs}$, $C_2 = C_3 = I$, $C_4 = -Z_{cm}$. In the figure, the blocks Z_m and Z_s represent the master and the slave dynamics (impedances), respectively, while C_m and C_s are the local master and slave controllers, f_h^* is an external force applied by the user, and f_e^* an exogenous force from the environment.

Summary and Future Directions

In these notes, an overview on telemanipulation has been presented with the aim of giving a general presentation of the impact of this area of robotics on both industry and research, of outlining typical problems encountered in dealing with remote manipulation systems, and of illustrating some approaches for their solution.

In this respect, it has to be pointed out that, besides the control schemes considered in these notes (purposely developed for telerobotic systems), many other schemes have been presented in the literature (see, e.g., Hokayem and Spong 2006, Ferre et al. 2007, and Arcara and Melchiorri 2002). More in general, however, a relevant literature exists, and important results have been presented from a methodological point of

Robot Teleoperation, Fig. 6 Block scheme of the four-channel architecture. (**a**) The human operator. (**b**) Master controller. (**c**) Communication line. (**d**) Slave controller. (**e**) Environment

view to face control problems of time-delay systems: see for example, Gu et al. (2003).

There are, however, other important aspects of telemanipulation which, for space constraints, can only be mentioned here, such as the "impedance shaping" (typical in applications in which there is a relevant dynamic/mechanical difference between the master and slave mechanisms) or criteria for defining (and measuring) performance of teleoperator systems, such as the "time to completion," criteria based on energy consumption, dexterity, and so on. Other interesting, and important, extensions are the possibility of controlling remote teams of robots cooperating for the execution of a common task (e.g., for aerial inspections, transport of heavy loads, etc.).

Future developments of robotic teleoperation systems will deal with the technological improvements of the user interface, giving to the operator more "realistic" feedback of the remote environment, the application of this type of technology to more complex situations, and the use of multi-robot systems controlled either by one or more cooperating users. Control will play in any case a fundamental role in these scenarios.

Cross-References

▶ Advanced Manipulation for Underwater Sampling
▶ Control of Linear Systems with Delays

▶ Disaster Response Robot
▶ Force Control in Robotics
▶ Model-Predictive Control in Practice
▶ Redundant Robots
▶ Robot Visual Control

Recommended Reading

Introductory and historical perspectives of telemanipulation, along with descriptions of several interesting applications of this technology, may be found in Ferre et al. (2007), Hokayem and Spong (2006), Sheridan (1992), and Vertut and Coiffet (1986). Specific applications, e.g., space, underwater, medical, and hazardous environment, are described in Duff et al. (2010), Hirzinger et al. (1992), http://marsrovers.jpl.nasa.gov/home/index.html, and Ridao et al. (2007). Some of the main control schemes specifically developed for this type of robotic devices are reported in Anderson and Spong (1989), Arcara and Melchiorri (2002), Colgate (1993), Hirche et al. (2007), Kim et al. (1992), and Niemeyer and Slotine (1991), while some basic background material on control theory is available in Gu et al. (2003) and Van der Schaft (2000).

Bibliography

Anderson JA, Spong MW (1989) Bilateral control of teleoperators with time delay. IEEE Trans Autom Control 34(5):494–501

Arcara P, Melchiorri C (2002) Control schemes for tele-operation with time delay: a comparative study. Int J Robot Auton Syst 38(1):49–64

Bejczy AK, Handlykken M (1981) Generalization of bilateral force-reflecting control of manipulators. In: Proceedings of the 4th Rom-An-Sy, Warsaw, pp 242–255

Bergamasco M, Frisoli A, Avizzano CA (2007) Exoskele-tons as man-machine interface systems for teleopera-tion and interaction in virtual environments. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) Advances in telerobotics. Springer, Berlin

Colgate JE (1993) Robust impedance shaping telemanip-ulation. IEEE Trans Robot Autom 9(4):374–384

Duff E, Caris C, Bonchis A, Taylor K, Gunn C, Adcock M (2010) The development of a telerobotic rock breaker. In: Howard A, Iagnemma K, Kelly A (eds) Field and service robotics. Springer tracts in advanced robotics, vol 62. Springer, Berlin/Heidelberg, pp 411–420

Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) (2007) Advances in telerobotics. Springer, Berlin

Ferrel WR (1966) Delayed force feedback. IEEE Trans Hum Factors Electr HFE-8:449–455

Goertz RC, Thompson RC (1954) Electronically con-trolled manipulators. Nucleonics 12(11): 46–47

Gu K, Kharitonov V, Chen J (2003) Stability of time-delay systems. Birkhauser, Boston

Hirche S, Ferre M, Barrio J, Melchiorri C, Buss M (2007) Bilateral control architectures for telerobotics. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) Advances in telerobotics. Springer, Berlin

Hirzinger G, Grunwald G, Brunner B, Heindl J (1992) A sensor-based telerobotic system for the space robot ex-periment ROTEX. In: Workshop on teleoperation and orbital robotics, 1992 IEEE international conference on robotics and automation, Nice

Hokayem PF, Spong MW (2006) Bilateral teleoperation: an historical survey. Automatica 42:2035–2057

http://marsrovers.jpl.nasa.gov/home/index.html

Kim WS, Hannaford B, Bejczy AK (1992) Force-reflecting and shared compliant control in operating telemanipulators with time delay. IEEE Trans Robot Autom 8(2):176–185

Kobrinskii A (1960) The thought control the machine: de-velopment of a bioelectric prosthesis. In: Proceedings of the 1st IFAC world congress on automatic control, Moscow

Lawrence DA, (1993) Stability and transparency in bi-lateral teleoperation. IEEE Trans Robot Autom 9(5): 624–637

Martin HL, Kuban DP (1985) Teleoperated robotics in hostile environments. Robotics International of SME, Dearborn

Niemeyer G, Slotine JJE (1991) Stable adaptive teleoper-ation. IEEE J Ocean Eng 16(1):152–162

Ridao P, Carreras M, Hernandez E, Palomeras N (2007) Underwater telerobotics for collaborative research. In: Ferre M, Buss M, Aracil R, Melchiorri C, Balaguer C (eds) Advances in telerobotics. Springer, Berlin

Sheridan TB (1992) Telerobotics, automation, and human supervisory control. MIT, Cambridge

Smith FM, Backman DK, Jacobsen SC (1992) Telerobotic manipulator for hazardous environments. J Robot Syst 9(2):251–260

Van der Schaft AJ (2000) L2-gain and passivity techniques in nonlinear control. Springer communications and control engineering series, 2nd edn. Springer, London

Vertut J, Coiffet P (1986) Robot technology, vol 3A: teleoperation and robotics: evolution and development. Prentice-Hall

Robot Visual Control

François Chaumette
Inria, Rennes, France

Abstract

This article presents the basic concepts of vision-based control, that is, the use of visual data to control the motions of a robotics system. It details the modeling steps allowing the design of kinematics control schemes. Applications are also described.

Keywords

Jacobian; Kinematics; Robot control; Visual ser-voing

Introduction

Visual control, also named visual servoing, refers to the use of computer vision data as input of real-time closed-loop control schemes to control the motion of a dynamic system, a robot typically (Chaumette and Hutchinson 2008; Hutchinson et al. 1996). It can be seen as sensor-based control from a vision sensor and relies on techniques from image processing, computer vision, and control theory.

An iteration of the control scheme consists of the following steps:
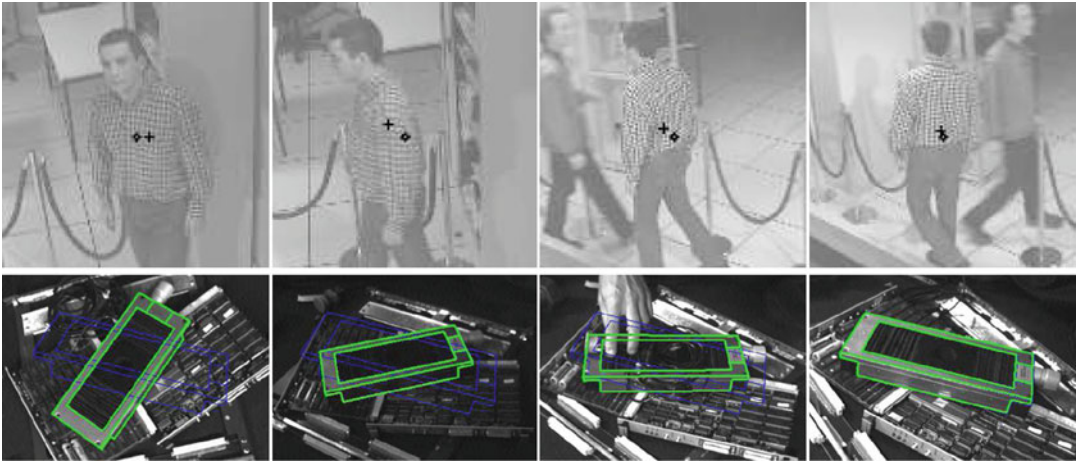
Robot Visual Control, Fig. 1 A few images acquired during two visual servoing tasks: on the *top*, pedestrian tracking using a pan-tilt camera; on the *bottom*, control-ling the 6 degrees of freedom of an eye-in-hand system so that an object appears at a particular position in the image (shown in *blue*)

- Acquire an image.
- Extract some useful image measurements.
- Compute the current value of the visual features used as inputs of the control scheme.
- Compute the error between the current and the desired values of the visual features.
- Update the control outputs, which are usually the robot velocity, to regulate that error to zero, i.e., to minimize its norm.

For instance, for the first example depicted on Fig. 1, the image processing part consists in extracting and tracking the center of gravity of the moving people, the visual features are composed of the two Cartesian coordinates of this center of gravity, and the control schemes compute the pan and tilt velocities so that the center of gravity is as near as possible of the image center despite the unknown motion of the people. In the second example where a camera mounted on a six-degrees-of-freedom robot arm is considered, the image measurements are a set of segments that are tracked in the image sequence. From these measurements and the knowledge of the 3D object model, the pose from the camera to the object is estimated and used as visual features. The control scheme now computes the six components of the robot velocity so that this pose reaches a particular desired value corresponding

to the object position depicted in blue on the images.

Basic Theory

Main if not all visual servoing tasks can be expressed as the regulation to zero of an error $\mathbf{e}(t)$ which is defined by

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a}) - \mathbf{s}^*(t). \qquad (1)$$

The parameters in (1) are defined as follows (Chaumette and Hutchinson 2008). The vector $\mathbf{m}(\mathbf{r}(t))$ is a set of image measurements (e.g., the image coordinates of interest points, or the area, the center of gravity, and other geometric characteristics of an object). These image measurements depend on the pose $\mathbf{r}(t)$ between the camera and the environment. They are used to compute a vector $\mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a})$ of visual features, in which $\mathbf{a}$ is a set of parameters that represent potential additional knowledge about the system (e.g., coarse camera intrinsic parameters or 3D model of objects). The vector $\mathbf{s}^*(t)$ contains the desired value of the features, which can be either constant in the case of a fixed goal or

R

varying if the task consists in following a specified trajectory.

Visual servoing schemes mainly differ in the way that the visual features are designed. As represented on Fig. 2, the two most classical approaches are named image-based visual servoing (IBVS), in which **s** consists of a set of 2D parameters that are directly expressed in the image (Espiau et al. 1992; Weiss et al. 1987), and pose-based visual servoing (PBVS), in which **s** consists of a set of 3D parameters related to the pose between the camera and the target (Weiss et al. 1987; Wilson et al. 1996). In that case, the 3D parameters have to be estimated from the image measurements either through a pose estimation process using the knowledge of the 3D target model, or through a partial pose estimation process using the properties of the epipolar geometry between the current and the desired images, or finally through a triangulation process if a stereovision system is considered. Inside IBVS and PBVS approaches, many possibilities exist depending on the choice of the features. Each choice will induce a particular behavior of the system. There also exist hybrid approaches, named 2-1/2D visual servoing, which combine 2D and 3D parameters in **s** in order to benefit from the advantages of IBVS and PBVS while avoiding their respective drawbacks (Malis et al. 1999).

The design of the control scheme is based on the link between the time variation of the features and the robot control inputs, which are usually the velocity of the robot joints **q**. This relation is given by

$$\dot{\mathbf{s}} = \mathbf{J_s}\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \qquad (2)$$

where $\mathbf{J_s}$ is the features Jacobian matrix, defined from the equation above similarly as the classical robot Jacobian. For an eye-in-hand system (see the left part of Fig. 3), the term $\frac{\partial \mathbf{s}}{\partial t}$ represents the time variation of **s** due to a potential object motion, while for an eye-to-hand system (see the right part of Fig. 3) it represents the time variation of **s** due to a potential sensor motion.

As for the features Jacobian, in the eye-in-hand configuration, it can be decomposed as Chaumette and Hutchinson (2008)
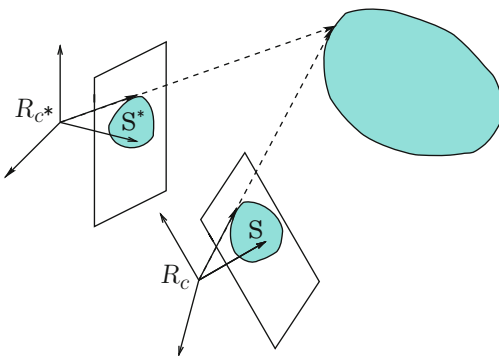
$$\mathbf{J_s} = \mathbf{L_s}\,{}^c\mathbf{V}_n\,\mathbf{J}(\mathbf{q}) \qquad (3)$$
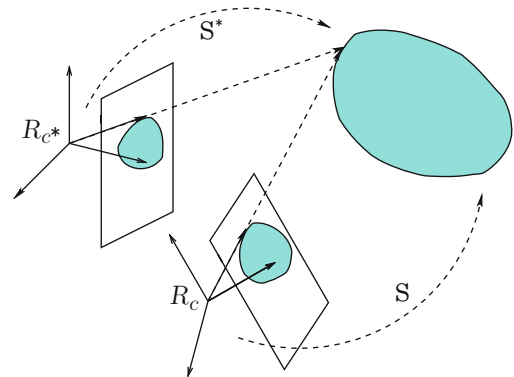
where

- $\mathbf{J}(\mathbf{q})$ is the robot Jacobian such that $\mathbf{v}_n = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{v}_n$ is the robot end effector velocity.
- ${}^c\mathbf{V}_n$ is the spatial motion transform matrix from the vision sensor to the end effector. It is given by

$$ {}^c\mathbf{V}_n = \begin{bmatrix} {}^c\mathbf{R}_n & [{}^c\mathbf{t}_n]_\times\,{}^c\mathbf{R}_n \\ \mathbf{0} & {}^c\mathbf{R}_n \end{bmatrix} \qquad (4)$$

where ${}^c\mathbf{R}_n$ and ${}^c\mathbf{t}_n$ are respectively, the rotation matrix and the translation vector between


Robot Visual Control, Fig. 2 If the goal is to move the camera from frame R_c to the desired frame R_{c*}, two main approaches are possible: IBVS on the *left*, where the features **s** and $\mathbf{s}^*$ are expressed in the image, and PBVS on the *right*, where the features **s** and $\mathbf{s}^*$ are related to the pose between the camera and the observed object

Robot Visual Control,
Fig. 3 In visual servoing, the vision sensor can be either mounted on the robot (eye-in-hand configuration) or remote and observing the robot (eye-to-hand configuration). For the same robot motion, the motion produced in the image will be opposite from one configuration to the other
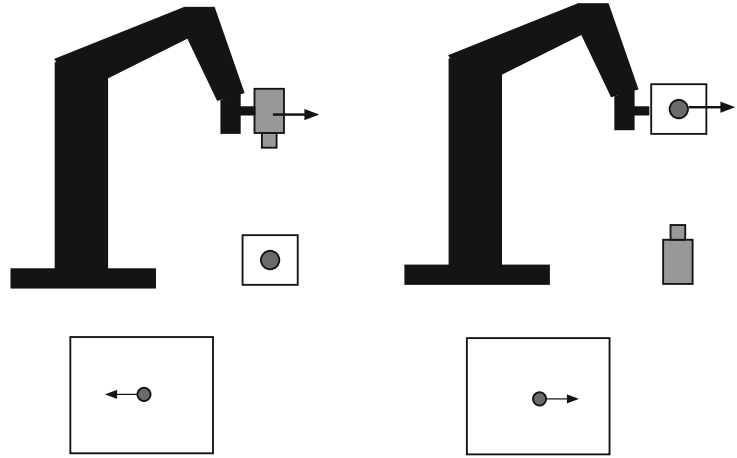
the sensor frame and the end effector frame and where $[^c\mathbf{t}_n]_\times$ is the skew symmetric matrix associated to $^c\mathbf{t}_n$. Matrix $^c\mathbf{V}_n$ is constant when the vision sensor is rigidly attached to the end effector, which is usually the case. Thanks to the robustness of closed-loop control schemes, a coarse approximation of $^c\mathbf{R}_n$ and $^c\mathbf{t}_n$ is sufficient in practice to get an estimation of $^c\mathbf{V}_n$. If needed, an accurate estimation is possible through classical hand-eye calibration methods.

• $\mathbf{L_s}$ is the interaction matrix of $\mathbf{s}$ defined such that $\mathbf{s} = \mathbf{L_s v}$ where $\mathbf{v}$ is the relative velocity between the camera and the environment.

In the eye-to-hand configuration, the features Jacobian $\mathbf{J_s}$ is composed of Chaumette and Hutchinson (2008)

$$\mathbf{J_s} = -\mathbf{L_s} \, ^c\mathbf{V}_f \, ^f\mathbf{V}_n \, \mathbf{J(q)} \qquad (5)$$

where

• $^f\mathbf{V}_n$ is the spatial motion transform matrix from the robot reference frame to the end effector frame. It is known from the robot kinematics model.
• $^c\mathbf{V}_f$ is the spatial motion transform matrix from the camera frame to the reference frame. It is constant as long as the camera does not move. In that case, similarly as for the eye-in-hand configuration, a coarse approximation of $^c\mathbf{R}_f$ and $^c\mathbf{t}_f$ is usually sufficient to get an estimation of $^c\mathbf{V}_f$.

A lot of works have concerned the modeling of the visual features and the determination of the analytical form of the interaction matrix. To give just an example, in the case of an image point with normalized Cartesian coordinates $\mathbf{x} = (x, y)$ and whose 3D corresponding point has depth Z, its interaction matrix is given by Espiau et al. (1992)

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \qquad (6)$$

where the three first columns contain the elements related to the three components of the translational velocity and where the three last columns contain the elements related to the three components of the rotational velocity.

By just changing the parameters representing the same image point, that is, by using the cylindrical coordinates defined by $\boldsymbol{\gamma} = (\rho, \theta)$ with $\rho = \sqrt{x^2 + y^2}$ and $\theta = \text{Arctan}(y/x)$, the interaction matrix of these parameters has a completely different form (Chaumette and Hutchinson 2008):

$$\mathbf{L_\gamma} = \begin{bmatrix} -c/Z & -s/Z & \rho/Z & (1+\rho^2)s & -(1+\rho^2)c & 0 \\ s/(\rho Z) & -c/(\rho Z) & 0 & c/\rho & s/\rho & -1 \end{bmatrix} \qquad (7)$$

where $c = \cos\theta$ and $s = \sin\theta$. This implies that using the Cartesian coordinates or the cylindrical coordinates as visual features will induce a different behavior, that is, a different robot

trajectory and a different trajectory of the point in the image.

Currently, the analytical form of the interaction matrix is available for most classical geometrical primitives, such as segments, straight lines, ellipses, moments related to planar objects of any shape (Chaumette 2004), and also coordinates of 3D points and pose parameters. Methods also exist to estimate off-line or online a numerical value of the interaction matrix. Omnidirectional vision sensors, the coupling between a camera and structured light, and even 2D echographic probes have also been studied. A large variety of visual features is thus available for many vision sensors.

Once the modeling step has been performed, the design of the control scheme can be quite simple. The most classical control scheme has the following form (Chaumette and Hutchinson 2008):

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}_s}^+ (\mathbf{s} - \mathbf{s}^*) + \widehat{\mathbf{J}_s}^+ \frac{\partial \mathbf{s}^*}{\partial t} - \widehat{\mathbf{J}_s}^+ \widehat{\frac{\partial \mathbf{s}}{\partial t}} \quad (8)$$

where λ is a positivive gain tuning the rate of convergence of the system and $\widehat{\mathbf{J}_s}^+$ is the Moore-Penrose pseudo inverse of an approximation or an estimation of the features Jacobian. The exact value of all its elements is indeed generally unknown since it depends of the intrinsic and extrinsic camera parameters, as well as of some 3D parameters such as the depth of the point in Eqs. (6) and (7).

The second term of the control scheme anticipates for the variation of $\mathbf{s}^*$ in the case of a nonconstant desired value. The third term compensates as much as possible a possible target motion in the eye-in-hand case and a possible camera motion in the eye-to-hand case. They are both null in the case of a fixed desired value and a motionless target or camera. They try to remove the tracking error in the other cases.

Following the Lyapunov theory, the stability of the system can be studied (Chaumette and Hutchinson 2008). Generally, visual servoing schemes can be demonstrated to be locally asymptotically stable (i.e., the robot will converge if it starts from a local neighborhood of

the desired pose) if the errors introduced in $\widehat{\mathbf{J}_s}$ are not too strong. Some particular visual servoing schemes can be demonstrated to be globally asymptotically stable (i.e., the robot will converge whatever its initial pose) under similar conditions.

Finally, when the visual features do not constrain all the robot degrees of freedom, it is possible to combine the visual task with supplementary tasks such as, for instance, joint limits avoidance or the visibility constraint (to be sure that the target considered will always remain in the camera field of view). In that case, the redundancy framework can be applied and the error to be regulated to zero has the following form:

$$\mathbf{e} = \widehat{\mathbf{J}_s}^+ (\mathbf{s} - \mathbf{s}^*) + (\mathbf{I} - \widehat{\mathbf{J}_s}^+ \widehat{\mathbf{J}_s}) \, \mathbf{e}_2 \qquad (9)$$

where $(\mathbf{I} - \widehat{\mathbf{J}_s}^+ \widehat{\mathbf{J}_s})$ is a projection operator on the null space of the visual task so that the supplementary task $\mathbf{e}_2$ will be achieved at best under the constraint that the visual task is realized (Espiau et al. 1992). A similar control scheme to (8) is now given by

$$\dot{\mathbf{q}} = -\lambda \, \mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}}. \qquad (10)$$

This scheme has for instance been applied for the first example depicted on Fig. 4 where the rotational motion of the mobile robot is controlled by vision, while its translational motion is controlled by the odometry to move at a constant velocity.

Applications

Potential applications of visual servoing are numerous. It can be used as soon as a vision sensor is available and a task is assigned to a dynamic system to control its motion. A non-exhaustive list of examples is (see Fig. 4):

- The control of a pan-tilt-zoom camera, as illustrated in Fig. 1 for the pan-tilt case
- Grasping using a robot arm
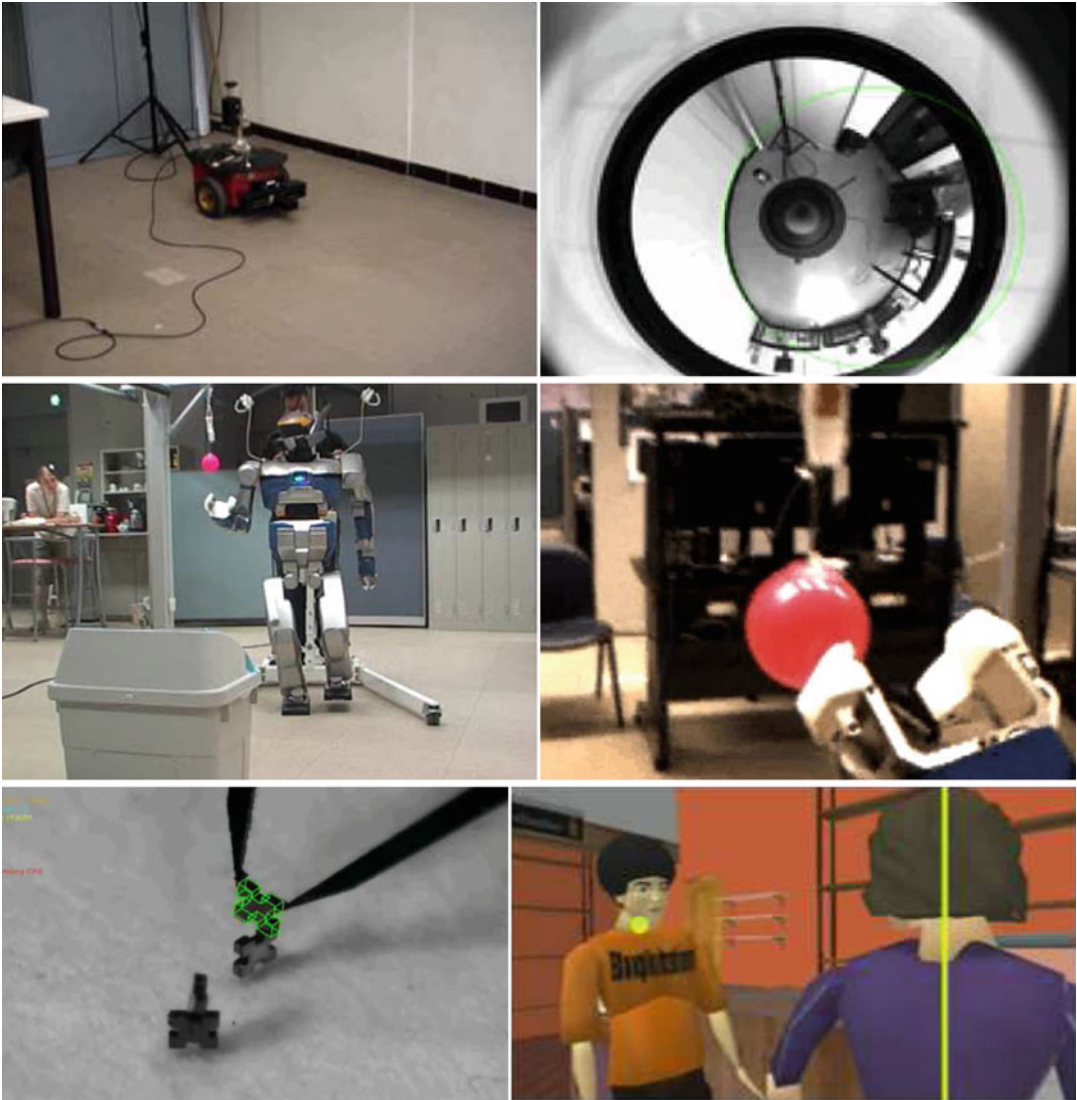
Robot Visual Control, Fig. 4 Few applications of visual servoing: navigation of a mobile robot to follow a wall using an omnidirectional vision sensor (*top row*), grasping a ball with a humanoid robot (*middle row*), assembly of MEMS and film of a dialogue within the constraints of a script in animation (*bottom row*)

- Locomotion and dexterous manipulation with a humanoid robot
- Micro- or nano-manipulation of MEMS or biological cells
- Pipe inspection by an underwater autonomous vehicle
- Autonomous navigation of a mobile robot in indoor or outdoor environment
- Aircraft landing
- Autonomous satellite rendezvous
- Biopsy using ultrasound probes or heart motion compensation in medical robotics
- Virtual cinematography in animation

Summary and Future Directions

Visual servoing is basically a nonlinear control problem. Several modeling works have been realized to design visual features so that the control

problem is transformed as much as possible to a linear control problem, leading to better stability properties. On one hand, improvements on this topic are still expected. On the other hand, the design of advanced control schemes, such as optimal control or model predictive control, is another way to make improvements. Then, taking into account dynamic constraints, such as non-holonomic constraints or underactuated systems, also necessitates the design of specific control laws.

Cross-References

▶ Lyapunov's Stability Theory
▶ Redundant Robots
▶ Robot Motion Control

Recommended Reading

In addition to the classical tutorial Hutchinson et al. (1996) and the most recent one Chaumette and Hutchinson (2008), the books Corke (1997, 2011) and the collection of papers Hashimoto (1993), Kriegman et al. (1998), and Chesi et al. (2010) provide a good overview of past and recent works in the field. The other references below cited in text present the main pioneering contributions in visual servoing.

Bibliography

Chaumette F (2004) Image moments: a general and useful set of features for visual servoing. IEEE Trans Robot 20(4):713–723

Chaumette F, Hutchinson S (2008) Visual servoing and visual tracking. In: Siciliano B, Khatib O (eds) Handbook of robotics, Chap. 24. Springer, Dordrecht, pp 563–583

Chesi G, Hashimoto K (eds) (2010) Visual servoing via advanced numerical methods. LNCIS, vol 401. Springer, Berlin

Corke P (1997) Visual control of robots: high-performance visual servoing. Wiley, New York

Corke P (2011) Robotics, vision and control. Springer tracts in advanced robotics, vol 73. Springer, Berlin

Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. IEEE Trans Robot Autom 8(3):313–326

Hashimoto K (ed) (1993) Visual servoing: real-time control of robot manipulators based on visual sensory feedback. World Scientific, Singapore

Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. IEEE Trans Robot Autom 12(5):651–670

Kriegman D, Hager G, Morse S (eds) (1998) The confluence of vision and control. LNCIS, vol 237. Springer, London

Malis E, Chaumette F, Boudet S (1999) 2-1/2D visual servoing. IEEE Trans Robot Autom 15(2):238–250

Weiss L, Sanderson A, Neuman C (1987) Dynamic sensor-based control of robots with visual feedback. IEEE J Robot Autom 3(5):404–417

Wilson W, Hulls C, Bell G (1996) Relative end-effector control using cartesian position-based visual servoing. IEEE Trans Robot Autom 12(5):684–696

Robust Adaptive Control

Anuradha M. Annaswamy
Active-adaptive Control Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Robust adaptive control pertains to the satisfactory behavior of adaptive control systems in the presence of nonparametric perturbations such as disturbances, unmodeled dynamics, and time delays. This article covers the highlights of robust adaptive controllers, methods used, and results obtained. Both methods of achieving robustness, which include modifications in the adaptive law and persistent excitation in the reference input, are presented. In both cases, results obtained for robustness to disturbances and unmodeled dynamics are discussed.

Keywords

Dead zone; Global boundedness; Parameter projection; Persistent excitation; Robustness; s-modification

Introduction

The central problem in adaptive control pertains to regulation and tracking of systems in the presence of parametric uncertainties. The classical adaptive control problem solved in 1980 assumed that the underlying transfer function had unknown parameters, but no other uncertainties. No disturbances, delays, time variations in parameters, or unmodeled dynamics were assumed to be present. Under these ideal conditions, it was shown that an adaptive controller can be designed so that the closed-loop system has bounded signals and that asymptotic regulation and tracking were possible.

With asymptotic regulation and tracking achieved under such ideal conditions, the goal of robust adaptive control was to ensure globally bounded signals in the closed-loop adaptive system when the plant was subjected to a variety of nonparametric perturbations such as external disturbances, time-varying parameters, unmodeled dynamics, and time delays. With adaptation in the control parameters in the ideal case accommodating parametric uncertainties, the approaches developed in robust adaptive control focused on developing solutions in the perturbed case to accommodate nonparametric uncertainties and improving on the classical adaptive controller which either underperformed or even exhibited instability with the introduction of nonparametric perturbations.

We briefly present the adaptive control solutions for the ideal case before proceeding with robust adaptive control.

Classical Adaptive Control

Adaptive Control of Plants with State Feedback

One of the very first problems where stable adaptive control was solved was for the case when states are accessible (Narendra and Kudva 1972), with the plant given by (The argument t is suppressed for the sake of convenience, except for emphasis.)

$$\dot{x}_p = A_p x_p + b\lambda u \qquad (1)$$

where $A_p \in \mathbb{R}^{n \times n}$ and the scalar λ are unknown parameters with b and the sign of λ known and (A_p, b) controllable. An adaptive controller that ensures global boundedness and asymptotic regulation and tracking for such plants is of the form

$$u = \theta_x^T(t)x_p + \theta_r(t)r, \qquad (2)$$

and the adaptive laws for adjusting the unknown parameters are given by

$$\dot{\theta} = -\text{sign}(\lambda)\Gamma\omega b_m^T Pe, \qquad (3)$$

where $\omega = \left[x_p^T, r\right]^T$ and $\theta = \left[\theta_x^T, \theta_r\right]^T$, x_m is the state of a reference model

$$\dot{x}_m = A_m x_m + br \qquad (4)$$

with A_m Hurwitz, and P being the solution of the Lyapunov equation $A_m^T P + PA_m = -Q$, $Q > 0$. The reference model in (4) is to be chosen so that certain matching conditions are satisfied, which are of the form

$$A_p + b\lambda\theta_x^{\star T} = A_m, \qquad \lambda\theta_r^* = 1 \qquad (5)$$

for some $\theta^* = [\theta_x^{*T}, \theta_r^*]^T$. In such a case, the controller in (2) and (3) guarantees stability and ensures that $x(t)$ tracks $x_m(t)$. The underlying Lyapunov function is quadratic in e and the parameter error $\tilde{\theta} = \theta - \theta^*$, given by

$$V = \frac{1}{2}\left(e^T Pe + \lambda\tilde{\theta}^T\Gamma^{-1}\tilde{\theta}\right) \qquad (6)$$

with a negative semi-definite time derivative $\dot{V}$ given by

$$\dot{V} \leq -e^T Qe. \qquad (7)$$

Adaptive Control of Plants with Output Feedback

Consider the *single-input single-output* (SISO) system of equations

$$y(t) = W(s)u(t) \qquad (8)$$

where $u \in \Re$ is the input, $y \in \Re$ the measurable output, and s the differential operator. The transfer function of the plant is parameterized as

$$W(s) \triangleq k_p \frac{Z(s)}{P(s)} \qquad (9)$$

where k_p is a scalar and $Z(s)$ and $P(s)$ are monic polynomials with $\deg(Z(s)) < \deg(P(s))$. The following assumptions will be made throughout:

Assumption 1 $W(s)$ is minimum phase.

Assumption 2 The sign of k_p is known.

Assumption 3 The relative degree n^* and the order of $W(s)$ are known.

The goal is to design a control input u so that the output y in (8) tracks the output y_m of the reference system

$$y_m(t) = W_m(s)r(t) \triangleq k_m \frac{Z_m(s)}{P_m(s)} r(t) \qquad (10)$$

where k_m is a scalar and $Z_m(s)$ and $P_m(s)$ are monic polynomials with $W_m(s)$ relative degree n^*.

The structure of the adaptive controller is now presented:

$$\dot{\omega}_1(t) = \Lambda\omega_1 + b_\lambda u(t) \qquad (11)$$

$$\dot{\omega}_2(t) = \Lambda\omega_2 + b_\lambda y(t) \qquad (12)$$

$$\omega(t) \triangleq [r(t), \; \omega_1^T(t), \; y(t), \; \omega_2^T(t)]^T \qquad (13)$$

$$\theta(t) \triangleq [k(t), \; \theta_1^T(t), \; \theta_0(t), \; \theta_2^T(t)]^T \qquad (14)$$

$$u = \theta^T(t)\omega \qquad (15)$$

where $\Lambda \in \Re^{(n-1)\times(n-1)}$ is Hurwitzx, $b_\lambda \in \Re^{n-1}$, $\omega_1, \omega_2 \in \Re^{n-1}$, and $\theta \in \Re^{2n}$ is an adaptive gain vector with $k(t) \in \Re$, $\theta_1(t) \in \Re^{n-1}$, $\theta_2(t) \in \Re^{n-1}$, and $\theta_0(t) \in \Re$.

The update law for the adaptive parameter differs depending on whether the relative degree of $W_m(s)$ is unity or greater than one and can be described as follows:

$$\dot{\theta}(t) = -\text{sign}(k_p)\Gamma e_y \omega \qquad n^* = 1 \qquad (16)$$

and

$$\dot{\theta}(t) = -\text{sign}(k_p)\Gamma \frac{e_a \zeta}{1 + \zeta^T\zeta} \qquad n^* \geq 2 \qquad (17)$$

where $e_y = y - y_m$, e_a is an augmented error, and ζ is a modified regressor, both of which are determined by the following equations:

$$\zeta = W(s)\omega, \quad \omega = [r, \omega_1^T, y, \omega_2^T]^T, \qquad (18)$$

$$e_2 = \theta^T\zeta - W(s)[\theta^T\omega] \qquad (19)$$

$$e_a = e_y + k_1(t)e_2 \qquad (20)$$

$$\dot{k}_1 = -\frac{e_a e_2}{1 + \zeta^T\zeta} \qquad (21)$$

The results of Narendra and Annaswamy (2005) guarantee that the above adaptive controller in Eqs. (11)–(21) will guarantee that $e_y(t)$ tends to zero as $t \to \infty$ with all signals remaining bounded in both the $n^* = 1$ and $n^* \geq 2$ cases.

Need for Robust Adaptive Control

When a disturbance η is present, the plant dynamics often is of the form

$$\dot{x}_p = A_p x_p + b\lambda(u + \eta(t)) \qquad (22)$$

while the reference model and the controller remain the same as in (4) and (2), respectively. This in turn necessitates new tools for the analysis and synthesis of adaptive systems. The main reason for this is the fact that the standard Lyapunov function candidate given by

$$V = \frac{1}{2}e^T P e + \frac{1}{2}\lambda\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \qquad (23)$$

together with the parameter adjustment as in (3) yields a time derivative

$$\dot{V} \leq -\frac{1}{2}e^T Q e + k_1\|e\|d_0 \qquad k_1 > 0, \quad (24)$$

where d_0 is an upper bound on the perturbation η. The second term on the right-hand side of (24)

causes $\dot{V}$ to be sign indefinite. This is because V is a function of both e and $\tilde{\theta}$, and therefore, the second term can be large compared to the first with the second argument of V, $\tilde{\theta}$, which can be arbitrary, causing $\dot{V}$ to be sign indefinite. The same property is what caused $\dot{V}$ to be semi-definite in the ideal case. Hence, in this perturbed case, no guarantees of boundedness can be provided. In fact, it can be shown that if $\eta(t)$ is chosen in a particular manner, the closed-loop signals can actually be shown to become unbounded, either in the presence of bounded disturbances (Narendra and Annaswamy 2005) or with unmodeled dynamics (Rohrs et al. 1985). This in turn led to the area of robust adaptive control.

Various approaches that have been developed under the rubric of robust adaptive control can be grouped into two categories. The first of these is related to modifications in the adaptive laws so as to ensure boundedness. These changes consist of modifications in the adaptive law (3) as

$$\dot{\theta} = -\Gamma \omega(t) b_m^T P e - \sigma g(\theta, e) \qquad (25)$$

The problem then reduces to finding a suitable $g(\theta, e)$. This is discussed in detail in the next sec-

tion. The second approach used in adaptive control pertains to the use of a persistently exciting reference signal r. The latter ensures parameter convergence of the adaptive system and therefore exponential stability. This in turn ensures robustness of the overall system. These details are addressed in section "Robust Adaptive Control with Persistently Exciting Reference Input."

Robust Adaptive Control with Modifications in the Adaptive Law

Robustness to Bounded Disturbances
When a bounded input disturbance η is present, the plant dynamics is changed as

$$\dot{x}_p = A_p x_p + b\lambda(u + \eta(t)), \qquad (26)$$

while the reference model and the controller remain the same as in (4) and (2), respectively. As mentioned above, a modification to the adaptive law as in (25) is needed. Over the years, different choices have been suggested for the nonlinear function $g(\theta, e)$. For example, these are chosen as

$$g(\theta, e) = \begin{cases} \theta & \text{Ioannou and Sun (2013)} \\ ||e||\theta & \text{Narendra and Annaswamy (2005)} \\ \theta \left(1 - \frac{||\theta||}{\theta_{\max}}\right)^2 & \text{Kreisselmeier and Narendra (1982)} \end{cases} \qquad (27)$$

where $\theta_{\max}$ is a known bound on the parameter θ. (One can choose to set σ to zero if $\|\theta\| \le \theta_{\max}$, as is done in Ioannou and Sun (2013), Tsakalis and Ioannou (1987) and many other references in the literature.) An alternate approach that is different from (25) is to not have an additive term $g(\cdot, \dot{})$ but rather set $\dot{\theta} = 0$ whenever the error e is small in some sense. Such a dead zone approach was suggested, for example, in Egardt (1979) and Peterson and Narendra (1982). It can be shown that each one of these choices leads to boundedness, which is described below. Without loss of generality, we assume that $\lambda > 0$.

With the same Lyapunov function candidate as in (23), its time derivative now becomes

$$\dot{V} \le -\frac{1}{2} e^T Q e + k_1 \|e\| \|\eta\|$$
$$\quad - \frac{1}{2} \|\tilde{\theta}\|^T g(\theta, e), \qquad k_1 > 0 \qquad (28)$$

The property of $g(.,.)$, together with the fact that η is bounded, ensures that $\dot{V} < 0$ outside a compact set Ω in the $(e, \tilde{\theta})$ space. This ensures global boundedness of both e and $\tilde{\theta}$. Boundedness of x_p follows.

In all of the above methods, the idea behind adding the term $g(e, \theta)$ is this: the parameter θ can drift away from the correct direction due to the term $k_1 \|e\| \|\eta\|$, and the construction of $g(e, \theta)$ is such that it counteracts this drift and keeps the parameter in check, by adding a negative quadratic term in $\tilde{\theta}$. The boundedness of both e and θ is simultaneously assured in the above since V has a time derivative $\dot{V}$ that is nonpositive outside a compact set in the $(e, \tilde{\theta})$ space. It should be noted however that this was possible to a large extent because η was bounded, and as a result, the sign-indefinite term remained linear in $\|e\|$.

An alternative procedure, originally proposed in Pomet and Praly (1992) and revised and refined in Khalil (2001) and Lavretsky (2010), proceeds in a slightly different manner. Here, the boundedness of θ is first established, independent of the error equation. It should be noted that a similar approach is adopted in the context of output feedback in plants with higher relative degree by using normalization and an augmented error approach (Narendra and Annaswamy 2005). In Khalil (2001) and Lavretsky (2010), no normalization is used but a projection algorithm. This is described below.

The projection algorithm for adjusting the parameter θ is given by

$$\dot{\theta} = \mathrm{Proj}(\theta, y), \qquad (29)$$

where

$$\mathrm{Proj}(\theta, y) = \begin{cases} y - \dfrac{\nabla f(\theta)(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2} y f(\theta) \\ \qquad \text{if } [f(\theta) > 0 \bigwedge y^T \nabla f(\theta) > 0] \\ y \qquad \text{otherwise} \end{cases} \qquad (30)$$

$$y = -e^T P b \omega \qquad (31)$$

$$f(\theta) = \frac{\| \theta \|^2 - \theta_{\max}'^2}{\varepsilon^2 + 2\varepsilon \theta_{\max}'} \qquad (32)$$

where $\theta_{\max}'$ and ε are arbitrary positive constants, and Ω_0 and Ω_1 are defined as

$$\begin{aligned} \Omega_0 &= \left\{ \theta \in \mathbb{R}^n | f(\theta) \leq 0 \right\} \\ \Omega_1 &= \left\{ \theta \in \mathbb{R}^n | f(\theta) \leq 1 \right\}. \end{aligned} \qquad (33)$$

From the above relations, one can show that

$$\theta(0) \in \Omega_0 \implies \theta(t) \in \Omega_1.$$

In addition,

$$\theta_{\max}' = \max_{\theta \in \Omega_0} (\|\theta\|), \quad \theta_{\max} = \max_{\theta \in \Omega_1} (\|\theta\|) \quad (34)$$

where $\theta_{\max} = \theta_{\max}' + \varepsilon$ (Matsutani et al. 2011).

Robustness to Unmodeled Dynamics

One of the major observations in the early eighties was the stark difference between the system signals in the ideal adaptive system and the perturbed adaptive system when the perturbation was due to a commonly present unmodeled dynamics such as those of an actuator used for control implementation. Among other references, the publication in Rohrs et al. (1985) pointed out the fact that when an adaptive controller prescribed for a first-order plant is evaluated with unmodeled dynamics present, instability occurs readily and for a wide range of command signals. A number of solutions have been suggested to alleviate this instability and form the subject matter of this section.

We consider the plant in (26) with an additional unmodeled dynamics so that

$$\begin{aligned} \dot{x}_p &= A_p x_p + b \lambda v \\ \dot{x}_\eta &= A_\eta x_\eta + b_\eta u, \quad v = \tilde{c}_\eta^T x_\eta. \end{aligned} \qquad (35)$$

where A_η is a Hurwitz matrix. If $\eta = v - u$, then the plant dynamics can be rewritten as

$$\dot{x}_p = A_p x_p + b\lambda(u + \eta) \tag{36}$$

Unlike the bounded disturbance case, no upper bound d_0 can be assumed to exist as η is a state-dependent disturbance. It is this that causes a huge difference between deriving boundedness in section "Robustness to Bounded Disturbances" and here in section "Robustness to Unmodeled Dynamics." Significant effort has been extended in the adaptive control community in this regard. These results fall into two categories (i) that assure global boundedness for a narrow class of unmodeled dynamics and (ii) that assure semi-global boundedness for a slightly larger class of unmodeled dynamics. More recently, some results have been obtained that are able to establish global boundedness with minimal restrictions on the unmodeled dynamics. In what follows, we give examples of each of the above two cases as well as the recent results.

Global Boundedness in the Presence of a Small Class of Unmodeled Dynamics

For the plant in (26), under assumptions in (5), the plant can be rewritten as

$$\dot{x} = A_m x + b\lambda(u + \theta_x^{*T} x + \eta) \tag{37}$$

where λ and θ_x^* are unknown, A_m and b are known, and $\eta = v - u$ whose state-space representation can be shown to be of the form

$$\dot{x}_\eta = A_\eta x_\eta + b_\eta u, \quad \eta = c_\eta^T x_\eta \tag{38}$$

for some vector c_η.

For a class of unmodeled dynamics $\{c_\eta, A_\eta, b_\eta\}$, if the control input in (2) and the projection algorithm in (29) with y and $f(\theta)$ chosen as in (31) and (32) are used, one can guarantee boundedness. In particular, if the inequality

$$k\theta_{x,\max} \lambda_{\max}\left(\frac{b_0}{\sigma_{A_\eta}}\right) < 1 \tag{39}$$

is satisfied, where b_0 is an upper bound on $||b_\eta||$ and σ_A denotes the singular value of the matrix A, then boundedness follows. That is, robustness of adaptive controllers can be ensured if the unmodeled dynamics is fast and their zeros are restricted in some sense.

A specific example of such an unmodeled dynamics is given by

$$c_\eta^T (sI - A_\eta)^{-1} b_\eta = \frac{-2\mu s}{1 + \mu s}. \tag{40}$$

Global Boundedness for a Large Class of Unmodeled Dynamics: A First-Order Example

A different approach can be taken for the problem of unmodeled dynamics which allows a global result, for a class of adaptive systems (Hussain et al. 2013). The main idea here is to use the projection algorithm and use properties of adaptive systems in conjunction with linear time-varying systems. This is presented in this section using a first-order plant.

We consider the control of

$$\dot{x}_p(t) = a_p x_p(t) + k_p v(t) \tag{41}$$

where a_p is unknown and k_p is known. It is assumed that $|a_p| \leq \bar{a}$, where $\bar{a}$ is a known positive constant. The unmodeled dynamics is given by (38) with

$$G_\eta(s) \triangleq c_\eta^T (sI_{n\times n} - A_\eta)^{-1} b_\eta. \tag{42}$$

The goal is to design the control input such that $x_p(t)$ follows $x_m(t)$ which is specified by the reference model

$$\dot{x}_m(t) = a_m x_m(t) + k_m r(t) \tag{43}$$

where $a_m < 0$ and $r(t)$ is the reference input. The adaptive controller we propose is given by

$$u(t) = \theta(t)x_p(t) + \frac{k_m}{k_p} r(t) \tag{44}$$

where the parameter $\theta(t)$ is updated using a projection algorithm given by

$$\dot{\theta}(t) = \gamma \,\mathrm{Proj}(\theta(t), -x_p(t)(x_p(t) - x_m(t))),$$
$$\gamma > 0 \qquad\qquad (45)$$

and

$$\mathrm{Proj}(\theta, y) = \begin{cases} \dfrac{\theta_{\max}^2 - \theta^2}{\theta_{\max}^2 - \theta_{\max}'^2}\, y & [\theta \in \Omega_A,\ y\theta > 0] \\[3em] y & \text{otherwise} \end{cases}$$
$$(46)$$

$$\Omega_0 = \{\theta \in \mathbb{R}^1 \mid -\theta_{\max}' \le \theta \le \theta_{\max}'\}$$
$$\Omega_1 = \{\theta \in \mathbb{R}^1 \mid -\theta_{\max} \le \theta \le \theta_{\max}\} \qquad (47)$$
$$\Omega_A = \Omega_1 \backslash \Omega_0$$

with positive constants $\theta_{\max}'$ and $\theta_{\max}$ given by

$$\theta_{\max}' > \frac{\bar{a} + |a_m|}{k_p}, \qquad \theta_{\max} = \theta_{\max}' + \varepsilon_0, \quad (48)$$

where ε_0 is an arbitrary constant. It can be shown that if $\theta_{\max}$ is chosen as in (48), then the closed

adaptive system specified by Eqs. (41)–(48) always has guaranteed bounded solutions for a class of unmodeled dynamics $G_\eta(s)$. There is an optimal value of ε_0, however, for which a largest class of $G_\eta(s)$ can be found.

It should be noted that in the Rohrs example in Rohrs et al. (1985), the plant is first order, with $a_p = -1$, and

$$G_\eta = \frac{w_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \qquad (49)$$

for $\zeta = 1$, $\omega_n = 15$. It is easy to show that for these values of ζ and ω_n, if $\theta_{\max} = 17$, then Eq. (48) is satisfied and that the closed-loop system is robust to G_η.

In general, for a first-order plant as in (41), it can be shown that the adaptive system is robust for G_η for all (ζ, ω_n) that satisfy the following inequalities for all $|a_p| \le \bar{a}$:

$$-a_p\zeta^2 + f(a_p, \omega_n)\zeta - \frac{k_p\theta_{\max}}{4} > 0$$
$$(50)$$
$$\omega_n > \omega_{n_{\min}}$$

where

$$f(a_p, \omega_n) = \frac{a_p^2 + \omega_n^2}{2\omega_n}$$

$$w_{n_{\min}} = \max\left(\frac{\bar{a}}{2\zeta}, 2\zeta\bar{a}, \sqrt{\bar{a}k_p\theta_{\max}}\left\{1 + \sqrt{1 - \frac{\bar{a}}{k_p\theta_{\max}}}\right\}\right) \qquad (51)$$

When a time delay τ is present in the plant to be controlled, the plant under consideration can be represented as in (37) where

$$\eta(t) = u(t - \tau) - u(t)$$

Similar results of global boundedness can be derived in this case as well (Matsutani 2013; Matsutani et al. 2012, 2013).

Robust Adaptive Control with Persistently Exciting Reference Input

We return to the plant in (26) with the control input as in (2) and the adaptive law as in (3).

When $\eta(t)$ is bounded with a finite upper bound d_0, it can be shown that no modifications are necessary in the adaptive law to ensure boundedness if the reference input is persistently exciting. It can be shown that if the reference input $r(t)$ is such that the vector ω^* defined as $\omega^* = [x_m^T, r]^T$ is persistently exciting with

$$\left| \frac{1}{T} \int_t^{t+T} \omega^{*T}(\tau)w\,d\tau \right| \ge kd_0 \quad \forall\, t \ge t_0, \forall w \in \mathbb{R}^n$$

where k, T are finite constants and w is a unit vector, then the adaptive system is well behaved,

i.e., has globally bounded solutions (Narendra and Annaswamy 2005).

An alternative approach for achieving robustness has been addressed in Anderson et al. (1986) that addresses local stability in the presence of persistently exciting signals. The starting point for this investigation is (35) but when all states are not accessible. Assuming that an output $y = c_p^T x_p$ is measurable and a controller as in (11)–(15) and a reference model as in (10) are used, the underlying error equation can be written as

$$e_1 = \overline{W}_m(s)\left(\tilde{\theta}^\top \omega + \overline{v}\right) \qquad (52)$$

where $\overline{W}_m(s)$ is asymptotically stable, $\tilde{\theta}$ is the parameter error vector, and $\overline{v}$ is the effect of the unmodeled dynamics η at the output. Suppose the standard adaptive law is used, and as a first step the perturbation $\overline{v}$ is ignored, the underlying error equation and the adaptive law are given by

$$e_1 = \overline{W}_m(s)\tilde{\theta}^\top \omega \qquad (53)$$

$$\dot{\tilde{\theta}} = -\mu e_1 \omega, \quad \mu > 0. \qquad (54)$$

If the origin in the $(e_1, \tilde{\theta})$ space of (53) and (54) is exponentially stable, all solutions of (52) are bounded for sufficiently small initial conditions and $\overline{v}(t)$. Therefore, the question that is of interest is the set of conditions of persistent excitation that will assure such an exponential stability. This is addressed in Anderson et al. (1986). The underlying tool is the Method of Averaging (Arnold 1982; Hale 1969; Krylov and Bogoliuboff 1943) used in the study of nonlinear oscillations and addresses the stability property of the differential equation

$$\dot{x} = \mu f(x, t, \mu), \qquad x(0) = x_0 \qquad (55)$$

where μ is a small parameter. By a process of averaging, the nonautonomous system in (55) is approximated by an autonomous differential equation in x_{av}, an averaged value of x. This autonomous system, which is easier to analyze, can be used to derive stability properties of (55).

In order to use the method of averaging for robust adaptive control, we write Eqs. (53) and (54) as

$$\begin{bmatrix} \dot{e} \\ \dot{\tilde{\theta}} \end{bmatrix} = \begin{bmatrix} A & b\omega^\top \\ -\mu\omega h^\top & 0 \end{bmatrix} \begin{bmatrix} e \\ \tilde{\theta} \end{bmatrix} \qquad (56)$$

Theorem 1 *Let $\omega(t)$ be bounded, almost periodic, and persistently exciting. Then there exists a $c^* > 0$ such that for all $\mu \in (0, c^*]$, the origin of (56) is exponentially stable if*

$$\Re\left[\lambda_i\left(\int_0^T \omega(t)\bar{W}_m(s)\omega^\top(t)dt\right)\right] > 0,$$
$$\forall i = 1, \ldots, n \qquad (57)$$

and is unstable if

$$\Re\left[\lambda_j\left(\int_0^T \omega(t)\bar{W}_m(s)\omega^\top(t)dt\right)\right] < 0,$$
$$\textit{for some } j = 1, \ldots, n \qquad (58)$$

In Kokotovic et al. (1985), it is further shown that $\omega(t)$ can be expressed at $\omega(t) = \sum_{k=-\infty}^{\infty} \Omega(iv_k)\exp(iv_k t)$ and the inequality in (57) can be satisfied if the condition

$$\sum_{k=-\infty}^{\infty} \Re\left[\bar{W}_m(iv_k)\right]\Re\left[\Omega(iv_k)\bar{\Omega}^\top(iv_k)\right] > 0 \qquad (59)$$

is satisfied, where $\bar{\Omega}(iv_k)$ is the complex conjugate of $\Omega(iv_k)$. Given a general transfer function $\bar{W}_m(s)$, there exists a large class of functions ω that satisfies (59), even when $\bar{W}_m(s)$ is not SPR.

ω in Theorem 1 is not an independent variable but rather an internal variable of the nonlinear system in (56). Hence, it cannot be shown to be bounded or persistently exciting. If ω_* represents the signal corresponding to ω in the reference model, it can be made to satisfy (57) by the proper choice of the reference input. Expressing $\omega = \omega_* + \omega_e$, ω will also be bounded, persistently exciting, and satisfy (57) if ω_e is small. This can be achieved by choosing the initial conditions $e(t_0)$ and $\tilde{\theta}(t_0)$ in (56) to be sufficiently small.

R

The conditions of Theorem 1 are then verified, and for a sufficiently small μ, exponential stability of the origin of (56) follows.

Theorem 1 provides conditions for exponential stability and instability when the solutions of the adaptive system are sufficiently close to the tuned solutions. These are very valuable in understanding the stability and instability mechanisms peculiar to adaptive control in the presence of different types of perturbations. Many of these results have been summarized and presented in a unified fashion in Anderson et al. (1986).

Cross-References

▶ Adaptive Control, Overview
▶ History of Adaptive Control
▶ Nonlinear Adaptive Control
▶ Stochastic Adaptive Control

Bibliography

Anderson BDO, Bitmead RR, Johnson CR Jr, Kokotovic PV, Kosut RL, Mareels IM, Praly L, Riedle BD (1986) Stability of adaptive systems: passivity and averaging analysis. MIT, Cambridge

Arnold VI (1982) Geometric methods in the theory of differential equations. Springer, New York

Egardt B (1979) Stability of adaptive controllers. Springer, New York

Hale JK (1969) Ordinary differential equations. Wiley–Interscience, New York

Hussain H, Matsutani M, Annaswamy A, Lavretsky E (2013) Adaptive control of scalar plants in the presence of unmodeled dynamics. In: 11th IFAC international workshop, ALCOSP, Caen, France, Jul 2013

Ioannou P, Sun J (2013) Robust adaptive control. Dover, Mineola

Khalil H (2001) Nonlinear systems, ch. 14.5. Prentice Hall, Upper Saddle River

Kokotovic P, Riedle B, Praly L (1985) On a stability criterion for continuous slow adaptation. Syst Control Lett 6:7–14

Kreisselmeier G, Narendra KS (1982) Stable model reference adaptive control in the presence of bounded disturbances. IEEE Trans Autom Control 27:1169–1175

Krylov AN, Bogoliuboff NN (1943) Introduction to nonlinear mechanics. Princeton University Press, Princeton

Lavretsky E (2010) Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. IEEE Trans Autom Control 57:1587–1591

Matsutani M (2013) Robust adaptive flight control systems in the presence of time delay. Ph.D. dissertation, Massachusetts Institute of Technology

Matsutani M, Annaswamy A, Gibson T, Lavretsky E (2011) Adaptive systems with guaranteed delay margins. In: 50th IEEE conference on decision and control and European control conference, Orlando, FL

Matsutani M, Annaswamy A, Lavretsky E (2012) Guaranteed delay margins for adaptive control of scalar plants. In: 2012 IEEE 51st annual conference on decision and control (CDC), Maui, Hawaii, pp 7297–7302

Matsutani M, Annaswamy A, Lavretsky E (2013) Guaranteed delay margins for adaptive systems with state variables accessible. In: American control conference, Washington, DC

Narendra KS, Annaswamy AM (2005) A new adaptive law for robust adaptation without persistent excitation. IEEE Trans Autom Control, 32:134–145

Narendra KS, Annaswamy AM (2005) Stable adaptive systems. Dover, Mineola

Narendra KS, Kudva P (1972) Stable adaptive schemes for system identification and control – parts I & II. IEEE Trans Syst Man Cybern 4:542–560

Peterson B, Narendra K (1982) Bounded error adaptive control. IEEE Trans Autom Control 27(6):1161–1169

Pomet J, Praly L (1992) Adaptive nonlinear regulation: estimation from the Lyapunov equation. IEEE Trans Autom Control 37(6):729–740

Rohrs C, Valavani L, Athans M, Stein G (1985) Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. IEEE Trans Autom Control 30(9):881–889

Tsakalis K, Ioannou P (1987) Adaptive control of linear time-varying plants. Automatica 23(4):459–468

Robust Control in Gap Metric

Li Qiu
Hong Kong University of Science and Technology, Hong Kong SAR, China

Abstract

Robust control needs to start with a model of system uncertainty. What is a good uncertainty model? First it needs to capture the possible system perturbations and uncertainties. Second it needs to be mathematically tractable. The gap metric was introduced by Zames and El-Sakkary for this purpose. Its study climaxed in an award-winning paper by Georgiou and Smith. A modified gap, called the ν-gap, was later discovered by

Vinnicombe and was shown to have advantages. With these metrics in hand, robust stabilization issues can be nicely addressed.

Keywords

Gap metric; H-infinity control; ν-gap metric; Robust stabilization; Uncertain system

Introduction

The gap is rooted in mathematical literature for the purpose of measuring the distance between unbounded operators (Kato 1976). It is introduced to control theory by Zames and El-Sakkary (1980) to measure the distance between systems and subsequently to model an uncertain system, with the recognition that a possibly unstable system is simply a possibly unbounded operator. Here only continuous-time systems will be treated. Discrete-time systems can be treated in an analogous way. Let us identify a linear time-invariant (LTI) system with its transfer function. The set of m-input p-output finite-dimensional LTI systems is then identified with the set $\mathcal{R}^{p \times m}$ of $p \times m$ real rational matrices. Such a system can be considered as a linear operator from input space $\mathcal{H}_2^m$ to output space $\mathcal{H}_2^p$, defined by the input-output relation $y = Pu$. Here $\mathcal{H}_2$ is the collection of all bounded-energy signals $x(s)$ satisfying

$$\|x\|_2 := \sup_{\sigma > 0} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |x(\sigma + j\omega)|^2 d\omega \right)^{1/2}$$
$$< \infty.$$

This operator is possibly unbounded since for an input $u \in \mathcal{H}_2^m$, the corresponding output $y = Pu$ is not necessarily in $\mathcal{H}_2^p$. It is bounded if and only if P is stable, i.e., if and only if $P \in \mathcal{RH}_\infty^{p \times m}$, the set of $p \times m$ real rational matrices bounded over $\mathrm{Re}\, s > 0$. In this case, the induced operator norm is the $\mathcal{H}_\infty$ norm of P:

$$\|P\|_\infty = \sup_{\mathrm{Re}\, s > 0} \bar{\sigma}[P(s)] = \sup_{\omega \in \mathbb{R}} \bar{\sigma}[P(j\omega)].$$

No matter whether or not P is stable, we define the graph of P as

$$\mathcal{G}_P = \left\{ \begin{bmatrix} u \\ y \end{bmatrix} \in \mathcal{H}_2^{m+p} : y = Pu \right\},$$

i.e., the graph is the set of all finite energy input-output pairs. It is easy to see that $\mathcal{G}_P$ is a linear subspace of $\mathcal{H}_2^{m+p}$ and a little more effort shows that it is closed. Hence it uniquely corresponds to a bounded linear operator on $\mathcal{H}_2^{m+p}$, called the orthogonal projection onto $\mathcal{G}_P$, denoted by $\Pi_{\mathcal{G}_P}$. Now with two systems $P_1, P_2 \in \mathcal{R}^{p \times m}$, the gap in between is defined as

$$\delta(P_1, P_2) = \|\Pi_{\mathcal{G}_{P_1}} - \Pi_{\mathcal{G}_{P_2}}\|.$$

That the gap is a metric in $\mathcal{R}^{p \times m}$ follows from the fact that the induced operator norm used above defines a metric on the set of all orthogonal projections.

With the gap between two systems, an uncertain system described by the gap is simply a gap metric ball with a center P, called the nominal system, and a radius r, qualifying the amount of uncertainty:

$$\mathcal{B}(P, r) = \{\tilde{P} \in \mathcal{R}^{p \times m} : \delta(\tilde{P}, P) < r\}.$$

Gap Computation and Robust Stabilization

With the basic definitions constructed above, the following questions are then asked:

Computation: How can the gap between two systems be computed?

Analysis: How much stability robustness does a stable feedback system have against gap uncertainty in the plant or in both the plant and the controller?

Synthesis: How can a feedback controller be designed so that the feedback system has optimal robustness against gap uncertainty?

For the question on computation, it is rather easy to see that if P_1 and P_2 are static, also said to be memoryless, systems, i.e., $P_1(s) = K_1$ and $P_2(s) = K_2$, then

R

$$\delta(P_1, P_2) = \bar{\sigma}[(I + K_1 K_1')^{-1/2}$$
$$(K_1 - K_2)(I + K_2' K_2)^{-1/2}].$$

In the single-input-single-output case, this is exactly the chordal distance between two numbers K_1 and K_2. Hence the expression above generalizes the chordal distance between two complex numbers to constant matrices. What if P_1 and P_2 are dynamic systems? It is not until Georgiou (1988) when the computation of the gap was settled by using the coprime factorization.

For each $P \in \mathcal{R}^{p \times m}$, there are

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}, \begin{bmatrix} M & U \\ N & V \end{bmatrix} \in \mathcal{RH}_\infty^{(m+p) \times (m+p)}$$

such that $P = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ and

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}.$$

These matrices are said to give a doubly coprime factorization of P. Also $P = NM^{-1}$ and $P = \tilde{M}^{-1}\tilde{N}$ are said to be right and left coprime factorizations, respectively. In the doubly coprime factorization, we can further require

$$M^T(-s)M(s) + N^T(-s)N(s) = I \text{ and}$$
$$\tilde{M}(s)\tilde{M}^T(s) + \tilde{M}(s)\tilde{M}^T(s) = I.$$

In this case, the coprime factorizations are said to be normalized.

Theorem 1 (Computation of the gap) *Let $P_i = N_i M_i^{-1}, i = 1, 2$, be normalized right coprime factorizations. Then*

$$\delta(P_1, P_2) = \max \left\{ \inf_{Q \in \mathcal{RH}_\infty^{m \times m}} \left\| \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} - \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} Q \right\|_\infty, \inf_{Q \in \mathcal{RH}_\infty^{m \times m}} \left\| \begin{bmatrix} M_2 \\ N_2 \end{bmatrix} - \begin{bmatrix} M_1 \\ N_1 \end{bmatrix} Q \right\|_\infty \right\}.$$

The problems of finding the two infima above are $\mathcal{H}_\infty$ model-matching problems, special forms of $\mathcal{H}_\infty$ control problems. See article ▸ Optimal Control via Factorization and Model Matching and article ▸ H-Infinity Control in this encyclopedia. In principle, they can be solved using the standard ways.

The analysis and synthesis questions are satisfactorily answered by Georgiou and Smith (1990). Let us consider the feedback system shown in Fig. 1.

Such a feedback system is denoted by a plant and controller pair or simply a feedback pair $(P, C) \in \mathcal{R}^{p \times m} \times \mathcal{R}^{m \times p}$. This closed-loop system is stable if the transfer matrix from $\begin{bmatrix} w_1 \\ -w_2 \end{bmatrix}$ to $\begin{bmatrix} u_1 \\ y_2 \end{bmatrix}$, nicknamed the Gang of Four matrix,

$$GoF = \begin{bmatrix} (I + PC)^{-1} & (I + PC)^{-1}C \\ C(I + PC)^{-1} & P(I + PC)^{-1}C \end{bmatrix}$$
$$= \begin{bmatrix} I \\ P \end{bmatrix}(I + PC)^{-1}[I \ C]$$

is stable, i.e., belongs to $\mathcal{RH}_\infty$.

Theorem 2 (Stability margin) *Assume (P, C) form a stable closed-loop system. All feedback systems $(\tilde{P}, C)$ with $\tilde{P} \in \mathcal{B}(P, r)$ are stable if and only if $r \leq \|GoF\|_\infty^{-1}$.*

It follows from Theorem 2 that $\|GoF\|_\infty^{-1}$ is the stability margin of the closed-loop system in Fig. 1. The natural design problem is then to design a controller C for a given P such that $\|GoF\|_\infty^{-1}$ is maximized or equivalently $\|GoF\|_\infty$ is minimized. Such a problem again is an $\mathcal{H}_\infty$ control problem, which is the topic of article ▸ H-Infinity Control in this encyclopedia. It is realized by Georgiou and Smith (1990) that this particular $\mathcal{H}_\infty$ control problem has some unique features. Let P have a normalized doubly coprime factorization and let

$$R(s) = M^T(-s)U(s) + N^T(-s)V(s).$$

Then
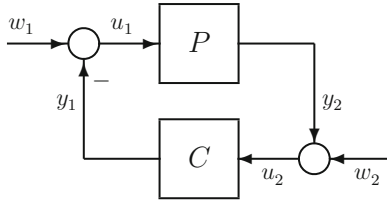
Robust Control in Gap Metric, Fig. 1 An uncertain feedback system

$$\inf_C \|GoF\|_\infty = \left(1 + \inf_{Q \in \mathcal{RH}_\infty^{m \times p}} \|R - Q\|_\infty\right)^{1/2}.$$

The minimization over Q above is a one-block $\mathcal{H}_\infty$ model-matching problem. It can be solved rather easily, much more easily than the $\mathcal{H}_\infty$ model-matching problem arising in the computation of gap. After finding an optimal Q, an optimal controller is obtained as

$$C = -(U - MQ)(V - NQ)^{-1}.$$

Qiu and Davison (1992a) extended Theorem 2 to the case when both the plant and controller are subject to uncertainty.

Theorem 3 (The arcsin theorem) *Assume (P, C) form a stable closed-loop system. All*

feedback systems $(\tilde{P}, \tilde{C}) \in \mathcal{B}(P, r_P) \times \mathcal{B}(C, r_C)$ are stable if and only if

$$\arcsin r_P + \arcsin r_C \leq \arcsin \|GoF\|_\infty^{-1}.$$

This theorem further strengthens the role of $\|GoF\|_\infty^{-1}$ as the stability robustness of the feedback system (P, C).

The ν-Gap

Partly because of the lack of efficient ways in computing the gap, there were efforts in seeking other metrics on $\mathcal{R}^{p \times m}$ with better numerical and analytical properties. Several such metrics were proposed, including the graph metric by Vidyasagar (1984), pointwise gap metric by Qiu and Davison (1992b), and ν-gap metric by Vinnicombe (1993). The winner is the ν-gap which is defined by ingeniously exploring the special structures and properties of rational matrices in $\mathcal{R}^{p \times m}$. For $P_1, P_2 \in \mathcal{R}^{p \times m}$, let $P_i = N_i M_i^{-1}, i = 1, 2$, be normalized right coprime factorizations. Define the ν-gap metric as

$$\delta_\nu(P_1, P_2) = \sup_{\omega \in \mathbb{R}} \bar{\sigma} \left\{[I + P_1(j\omega)P_1(j\omega)^*]^{-1/2}[P_1(j\omega) - P_2(j\omega)][I + P_2(j\omega)^* P_2(j\omega)]^{-1/2}\right\}$$

if $\det[M_2^T(-s)M_1(s) + N_2^T(-s)N_1(s)]$ has equal number of unstable poles and zeros and $\delta_\nu(P_1, P_2) = 1$ otherwise. Apparently ν-gap is easier to compute than the gap. When the pole-zero number condition is satisfied, the ν-gap is the peak of the chordal distance between the system frequency responses. The ν-gap is no greater than the gap, i.e.,

$$\delta_\nu(P_1, P_2) \leq \delta(P_1, P_2).$$

Hence the ν-gap ball

$$\mathcal{B}_\nu(P, r) = \{\tilde{P} \in \mathcal{R}^{p \times m} : \delta_\nu(\tilde{P}, P) < r\}$$

is a superset of the gap ball with the same center and radius. Theorems 2 and 3 can be restated

with the gap balls $\mathcal{B}$ replaced by the new gap balls $\mathcal{B}_\nu$. Consequently the restated Theorems 2 and 3 are less conservative than the original versions for the gap. The optimal robust stabilization problems for the gap and the ν-gap are the same: design C to maximizing $\|GoF\|_\infty^{-1}$ for a given P.

Summary and Future Directions

The gap, as well as the ν-gap, and the associated robust control theory can be extended to infinite dimensional systems as in Georgiou and Smith (1992) and Ball and Sasane (2012), time-varying systems as in Foias, Georgiou, and Smith et al.

(1993) and Feintuch (1998), and even nonlinear systems as in Georgiou and Smith (1997), Anderson et al. (2002), James et al. (2005), and Bian and French (2005), in varying degrees. There are still research opportunities in these extensions. The use of normalized coprime factorizations seems to be an obstacle in these extensions.

For a plant P, the controller optimizing $\|GoF\|_\infty$ is not always a good controller. This gives another example where "optimal" is not always equal to "good." One reason is that the actual plant uncertainty cannot necessarily be well described by a gap ball or a ν-gap ball. Another reason is that performance issues other than the stability robustness, such as tracking and disturbance rejection, are not taken into account in the optimization. The actual plant uncertainty might be better described by a gap ball centered at a frequency-shaped plant $\tilde{P} = W_o P W_i$ where W_i and W_o are real rational weighting matrices which can also be chosen to address tracking and disturbance rejection requirements. In this case, an optimal controller $\tilde{C}$ can then be designed to optimize the GoF matrix corresponding the shaped plant $\tilde{P}$. Finally $C = W_i \tilde{C} W_o$ is used as a designed controller for the original plant P. With the proper choice of W_i and W_o, it is more likely that a good controller will result. This loop-shaping design method was proposed in McFarlane and Glover (1992) and further developed in Vinnicombe (2001).

In the process of obtaining the arcsin theorem, it has been realized that the gap and even more so the ν-gap are closely related to the canonical angles between linear subspaces. In fact the gap is the sin of the largest canonical angle between certain subspaces and the largest canonical angle itself is also a metric, a better one in some geometric sense. For the latest development on canonical angles, see Qiu et al. (2008) and Zhang and Qiu (2010).

In addition to the effort in deepening and expanding the notion of gap and its use in robust control, there is also effort in making it more accessible and more closely related to classical frequency response analysis; see Qiu and Zhou (2013). It again appears that the use of coprime factorizations in the current theory is hindering this effort. Hence, circumventing the use of coprime factorizations, normalized or not, in the development of the gap would help its extension and popularization.

Cross-References

- ▶ H-Infinity Control
- ▶ Optimal Control via Factorization and Model Matching
- ▶ Robust Synthesis and Robustness Analysis Techniques and Tools

Recommended Reading

The most authoritative work on gap, ν-gap and the associated robust stabilization theory is the comprehensive monograph Vinnicombe (2001). This theory is inherently an input-output frequency domain theory. However many related computations, such as those of doubly normalized coprime factorizations, $\mathcal{H}_\infty$ model matching, and the optimal controller synthesis, can be done using state-space formulas and further using MATLAB programs. Vinnicombe (2001) contains a list of such state-space formulas. This theory provides a good example of the once popular and successful philosophy behind the linear multivariable control theory: thinking in terms of transfer functions and computing in term of state-space equations.

The system and control background needed to understand and study the gap, the ν-gap, and robust stabilization, in particular the coprime factorization and frequency domain stabilization theory, can be found in Vidyasagar (1985).

The book Zhou and Doyle (1998) also contains considerable content on gap based robust control.

Bibliography

Anderson BDO, Brinsmead TS, De Bruyne F (2002) The Vinnicombe metric for nonlinear operators. IEEE Trans Autom Control 47:1450–1465

Ball JA, Sasane AJ (2012) Extension of the ν-metric. Complex Anal Oper Theory 6:65–89

Bian W, French M (2005) Graph topologies, gap metrics, and robust stability for nonlinear systems. SIAM J Control Optim 44:418–443

El-Sakkary AK (1985) The gap metric: robustness of stabilization of feedback systems. IEEE Trans Autom Control 30:240–247

Feintuch A (1998) Robust control theory in Hilbert space. Springer, New York

Foias C, Georgiou TT, Smith MC (1993) Robust stability of feedback systems: a geometric approach using the gap metric. SIAM J Control Optim 31:1518–1537

Georgiou TT (1988) On the computation of the gap metric. Syst Control Lett 11:253–257

Georgiou TT, Smith MC (1990) Optimal robustness in the gap metric. IEEE Trans Autom Control 35:673–687

Georgiou TT, Smith MC (1992) Robust stabilization in the gap metric: controller design for distributed plants. IEEE Trans Autom Control 37:1133–1143

Georgiou TT, Smith MC (1997) Robustness analysis of nonlinear feedback systems: an input-output approach. IEEE Trans Autom Control 42:1200–1221

Glover K, McFarlane DC (1989) Robust stabilization of normalized coprime factor plant descriptions with $\mathcal{H}_\infty$ bounded uncertainties. IEEE Trans Autom Control 34:821–830

James MR, Smith MC, Vinnicombe G (2005) Gap metrics, representations and nonlinear robust stability. SIAM J Control Optim 43:1535–1582

Kato T (1976) Perturbation theory for linear operators, 2nd edn. Springer, Berlin

McFarlane DC, Glover K (1992) A loop shaping design procedure using $\mathcal{H}_\infty$-synthesis. IEEE Trans Autom Control 37:759–769

Qiu L, Davison EJ (1992a) Feedback stability under simultaneous gap metric uncertainties in plant and controller. Syst Control Lett 18:9–22

Qiu L, Davison EJ (1992b) Pointwise gap metrics on transfer matrices. IEEE Trans Autom Control 37:741–758

Qiu L, Zhang Y, Li CK (2008) Unitarily invariant metrics on the Grassmann space. SIAM J Matrix Anal 27:501–531

Qiu L, Zhou K (2013) Preclassical tools for postmodern control. IEEE Control Syst Mag 33(4): 26–38

Vidyasagar M (1984) The graph metric for unstable plants and robustness estimates for feedback stability. IEEE Trans Autom Control 29:403–418

Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge

Vinnicombe G (1993) Frequency domain uncertainty and the graph topology. IEEE Trans Autom Control 38:1371–1383

Vinnicombe G (2001) Uncertainty and feedback: $\mathcal{H}_\infty$ loop-shaping and the ν-gap metric. Imperial Collage Press, London

Zames G, El-Sakkary AK (1980) Unstable systems and feedback: the gap metric. In: Proceedings of the 16th Allerton conference, Illinois, pp 380–385

Zhang Y, Qiu L (2010) From subadditive inequalities of singular values to triangular inequalities of canonical angles. SIAM J Matrix Anal Appl 31:1606–1620

Zhou K, Doyle JC (1998) Essentials of robust control. Prentice Hall, Upper Saddle River

Robust Control of Infinite Dimensional Systems

Hitay Özbay
Department of Electrical and Electronics
Engineering, Bilkent University, Ankara, Turkey

Abstract

Basic robust control problems are studied for the feedback systems where the underlying plant model is infinite dimensional. The $\mathcal{H}_\infty$ optimal controller formula is given for the mixed sensitivity minimization problem with rational weights. Key steps of the numerical computations required to determine the controller parameters are illustrated with an example where the plant model include time delay terms.
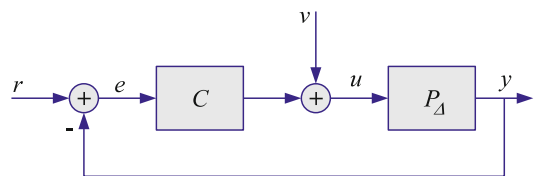
Keywords

Coprime factorizations; Direct design methods; Inner-outer factorizations

Introduction

Robust control deals with the feedback system shown in Fig. 1, where P_Δ represents the uncertain physical plant and C is a fixed controller to be designed.

Here, it is assumed that the controller and the plant are linear time invariant (LTI) systems and

Robust Control of Infinite Dimensional Systems, Fig. 1 Feedback system $\mathcal{F}(C, P_\Delta)$ with fixed controller C and uncertain plant P_Δ

they are represented by their transfer functions. Furthermore, P_Δ satisfies the following conditions:

$$P_\Delta(s) = P(s) + \Delta(s)$$

where P is the nominal plant model, with $P(s)$ and $P_\Delta(s)$ having the same number of poles in $\mathbb{C}_+$; and there is a known uncertainty bound $W(s)$ satisfying

$$|\Delta(j\omega)| < |W(j\omega)| \quad \forall\, \omega \in \mathbb{R}.$$

Definition 1 All P_Δ satisfying the above conditions are said to be in *the set of uncertain plants* $\mathcal{P}$, which is characterized by the given functions $P(s)$ and $W(s)$.

Depending on physical system modeling, other forms of uncertainty representations can be more convenient than the additive unstructured uncertainty model taken here; see, e.g., Doyle et al. (1992), Özbay (2000), and Zhou et al. (1996) for the examples of multiplicative, coprime factor, parametric, and structured uncertainty descriptions. Note that for notational convenience and simplicity of the presentation, single-input-single-output (SISO) plants are considered here; for extensions to multi-input-multi-output (MIMO) plants, see, e.g., Curtain and Zwart (1995).

When the plant under consideration is infinite dimensional, the transfer function $P(s)$ is irrational, i.e., it cannot be expressed as a ratio of two polynomials (it does not admit a finite-dimensional state-space representation). Typical examples of such systems are spatially distributed parameter systems modeled by partial differential equations, fractional-order systems, and systems with time delays. The reader is referred to Curtain and Morris (2009) for examples of transfer functions of distributed parameter systems. There are many interesting industrial applications where fractional-order transfer functions are used for modeling and control, see, e.g., Monje et al. (2010); typically, such functions are rational in s^α, where α is a rational number in the open interval $(0, 1)$. Transfer functions of systems with time delays involve terms like e^{-hs} where $h > 0$ is the delay; see Sipahi et al. (2011) for

various real-life examples where time-delay models appear. Transfer functions considered here are functions of the complex variable s with real coefficients, so $\overline{P(\bar{s})} = P(s)$ where $\bar{s}$ denotes the complex conjugate of s.

Definition 2 A linear time invariant system H is said to be *stable* if its transfer function $H(s)$ is bounded and analytic in $\mathbb{C}_+$. In this case, the *system norm* is

$$\|H\| = \|H\|_\infty = \sup_{\mathrm{Re}(s)>0} |H(s)|,$$

which is equivalent to the *energy amplification* through the system H; see Doyle et al. (1992) and Foias et al. (1996).

Definition 2 is sometimes called the $\mathcal{H}_\infty$-stability, and in this setting, the set of all stable plants is the function space $\mathcal{H}_\infty$. It is worth noting that for infinite-dimensional systems, there are other definitions of stability (Curtain and Zwart 1995; Desoer and Vidyasagar 2009), leading to different measures of the system norm.

Robust Control Design Objectives

Let $\mathcal{F}(C, P_\Delta)$ denote the feedback system shown in Fig. 1. This system is said to be *robustly stable* if all the transfer functions from external inputs (r, v) to internal signals (e, u) are in $\mathcal{H}_\infty$ for all $P_\Delta \in \mathcal{P}$. In the controller design, robust stability of the feedback system is the primary constraint.

The feedback system $\mathcal{F}(C, P_\Delta)$ is robustly stable if and only if the following conditions hold; see, e.g., Doyle et al. (1992) and Foias et al. (1996),

(a) $S, CS, PS \in \mathcal{H}_\infty$, where $S = (1 + PC)^{-1}$, and

(b) $\|W\,CS\|_\infty \le 1$.

In order to illustrate these design constraints for robustly stabilizing controller, as an example, consider a strictly proper stable plant, i.e.,

$$P \in \mathcal{H}_\infty \quad \text{with} \quad \lim_{|s|\to\infty} |P(s)| = 0.$$

In this case, all controllers in the form $C = Q/(1 - PQ)$ satisfy condition (a) for any $Q \in \mathcal{H}_\infty$ (moreover, any controller C satisfying (a) must be in this form for some $Q \in \mathcal{H}_\infty$). Now consider a rational $W(s)$ with a stable Q such that $|Q(j\omega)|$ is a continuous function of $\omega \in \mathbb{R}$. Then, condition (b) becomes

$$\|WQ\|_\infty \leq 1 \iff |Q(j\omega)| \leq |W(j\omega)|^{-1}$$
$$\forall \omega \in \mathbb{R}.$$

So, whenever the modeling uncertainty is "large" on a frequency band $\omega \in \Omega$, the magnitude of Q should be "small" in this region.

When the plant is unstable, say $p \in \mathbb{C}_+$ is a pole of $P(s)$ of multiplicity one, conditions (a) and (b) impose a restriction on the controller, that leads to

$$1 \geq \|\mathrm{WCS}\|_\infty \geq \left| \frac{W(p)}{N(p)} \right| \quad \text{where} \ N(p)$$
$$= \lim_{s \to p} \frac{(s - p)}{(s + \overline{p})} \ P(s).$$

So, a necessary condition, for (b) to hold in this case, is $|W(p)| \leq |N(p)|$, which means that the modeling uncertainty at the unstable pole of the plant should be small enough for the existence of a robustly stabilizing controller. This is one of the fundamental quantifiable limitations of feedback systems with unstable plants; see Stein (2003) for further discussions on other limitations.

Many other performance-related design objectives, such as reference tracking and disturbance attenuation, are captured by the *sensitivity minimization*, which is defined as finding a controller satisfying (a) and achieving

(c) $\|W_1 S\|_\infty \leq \gamma$

for the smallest possible $\gamma > 0$, for a given stable sensitivity weight $W_1(s)$. Selection of W_1 depends on the class of reference signals and disturbances considered; see Doyle et al. (1992), Özbay (2000), and Stein (2003) for general guidelines. Stability robustness and performance objectives defined above can be blended to define a single

$\mathcal{H}_\infty$-optimization problem, known as the *mixed sensitivity minimization*: given W_1, W_2, P, find a controller C satisfying (a) and achieving

(d) $\left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty :=$

$$\sup_{\mathrm{Re}(s) > 0} \left(|W_1(s)S(s)|^2 + |W_2(s)T(s)|^2 \right)^{\frac{1}{2}} \leq \gamma$$

for the smallest possible $\gamma > 0$, where $T(s) := 1 - S(s)$ and $W_2(s)$ represents the multiplicative uncertainty bound, with $|W_2(j\omega)| = |W(j\omega)|/|P(j\omega)|$, $\forall \ \omega \in \mathbb{R}$. The smallest achievable γ is the optimal performance level γ_{opt} and the corresponding controller is denoted by C_{opt}. Typically, when P is infinite dimensional so is the optimal controller.

Design Methods

Approximation of the Plant
One possible way to design a robust controller for an infinite-dimensional plant P is to design a robust controller C_a for an approximate finite-dimensional plant P_a; (for a frequency domain approximation technique for infinite-dimensional systems, see Gu et al. 1989). When W_1, W_2, and P_a are finite dimensional, standard state-space methods, Zhou et al. (1996), can be used to find an $\mathcal{H}_\infty$ controller C_a achieving

$$\left\| \begin{bmatrix} W_1 S_a \\ W_2 T_a \end{bmatrix} \right\|_\infty \leq \gamma_a$$

for the smallest possible γ_a, where $S_a := (1 + P_a C_a)^{-1}$ and $T_a = (1 - S_a)$. Then, the controller $C = C_a$ satisfies (a) and achieves the performance objective (d) with

$$\gamma = (\gamma_a + \varepsilon) \ \frac{1}{1 - \varepsilon}, \quad \varepsilon := \|C_a S_a (P - P_a)\|_\infty,$$

where it is assumed that the approximation of the plant is made in such a way that $\varepsilon < 1$. Clearly, if $\gamma_a \to \gamma_{\mathrm{opt}}$ as $\varepsilon \to 0$, then $\gamma \to \gamma_{\mathrm{opt}}$ as

R

$\varepsilon \to 0$. The conditions under which $\gamma_a \to \gamma_{\mathrm{opt}}$ are discussed in Morris (2001).

Direct Design Methods

The classical two-Riccati equation approach, Zhou et al. (1996), developed for finite-dimensional systems, has been extended to various classes of infinite-dimensional systems by using the state-space techniques where semigroup theory plays an important role; see van Keulen (1993) for further details.

In order to illustrate some of the key steps of a frequency domain method developed in Foias et al. (1996), consider a specific example where the plant is given as

$$P(s) = \frac{(s-1)(s+2)\,e^{-hs}}{(s^2+2s+2)(s+1-3e^{-2hs})} \, ,$$
$$h = \ln(2) \approx 0.693. \qquad (1)$$

First, compute the location of the poles in $\mathbb{C}_+$ using available numerical tools for finding the roots of quasi-polynomials; see, e.g., Sipahi et al. (2011) for references. For the simple example chosen here, $P(s)$ has only one pole in $\mathbb{C}_+$, at $s = 0.5$ (for larger values of h, the number of unstable poles of P may be higher). Now, the plant can be factored as follows:

$$P(s) = \frac{M_N(s)N_O(s)}{M_D(s)} \qquad (2)$$

where

$$M_N(s) = \frac{s-1}{s+1}\,e^{-hs} \qquad M_D(s) = \frac{s-0.5}{s+0.5}$$

are all-pass (inner) transfer functions and

$$N_O(s) = \frac{(s+2)(s+1)}{(s^2+2s+2)(s+0.5)}$$
$$\left(\frac{s-0.5}{s+1-3e^{-2hs}} \right)$$

is a minimum-phase (outer) transfer function. Note that

$$\frac{s-0.5}{s+1-3e^{-2hs}} = \frac{1}{1+H_F(s)} \, ,$$
$$H_F(s) = 1.5\,\frac{1-e^{-2h(s-0.5)}}{s-0.5}. \qquad (3)$$

The impulse response of H_F is $h_F(t) = 1.5e^{t/2}$ when $t \in [0, 2h]$ and $h_F(t) = 0$ otherwise. Stability of N_O can also be verified from the Nyquist graph of H_F. Also, note that $N_O(s)$ can be factored as $N_O(s) = N_1(s)N_2(s)$ where

$$N_1(s) = \frac{(s+2)(s+1)}{(s^2+2s^2+2)} \left(\frac{1}{1+H_F(s)} \right),$$
$$N_2(s) = \frac{1}{s+0.5}$$

with $N_1, N_1^{-1} \in \mathcal{H}_\infty$ and N_2 is finite-dimensional (first order in this example).

The above steps illustrate *coprime factorizations* and *inner-outer factorizations* for systems with time delays (retarded case). For systems represented by PDEs or integrodifferential equations, plant transfer function can be factored similarly, provided that the poles and zeros in $\mathbb{C}_+$ can be computed numerically.

When the plant is in the form (2) given above and the weights W_1 and W_2 are rational, the optimal performance level and the corresponding optimal controller is obtained by the following procedure (see Foias et al. (1996) for details).

- *Controller parameterization* transforms the mixed sensitivity minimization to a problem of finding the smallest $\gamma > 0$ for which there exists $Q \in \mathcal{H}_\infty$ such that

$$\left\| \begin{bmatrix} W_1 \\ 0 \end{bmatrix} - \begin{bmatrix} W_1 N_2 \\ -W_2 N_2 \end{bmatrix} M_N(R + M_D Q) \right\|_\infty \le \gamma$$

where $R(s)$ is a rational function (whose order is one less than the order of M_D) satisfying certain interpolation conditions at the zeros of $M_D(s)$.

- A *spectral factorization* determines $W_0 \in \mathcal{H}_\infty$ such that $W_0^{-1} \in \mathcal{H}_\infty$ and

$$\left(|W_1(j\omega)|^2 + |W_2(j\omega)|^2 \right) |N_2(j\omega)|^2$$
$$= |W_0(j\omega)|^2 \quad \forall \, \omega \in \mathbb{R},$$

(here, it is assumed that $W_2 N_2$ and $(W_2 N_2)^{-1}$ are in $\mathcal{H}_\infty$).

- By using the norm preserving property of the unitary matrices and the *commutant lifting theorem*, it has been shown that

$$\gamma_{\mathrm{opt}} = \left\| \begin{bmatrix} \Gamma \\ \Upsilon \end{bmatrix} \right\|$$

where Γ is the *Hankel operator* whose symbol is

$$M_D(-s)\left(M_N(-s) W_0^{-1}(-s) N_2(-s) W_1(-s) W_1(s) - W_0(s) R(s) \right)$$

and Υ is the *Toeplitz operator* whose symbol is $W_1(s) W_2(s) N_2(s) W_0^{-1}(s)$. Moreover, under mild technical assumptions, the optimal controller is obtained from a nonzero $\psi_o \in \mathcal{H}_2$ satisfying

$$\left(\gamma_{\mathrm{opt}}^2 - (\Gamma^* \Gamma + \Upsilon^* \Upsilon) \right) \psi_o = 0$$

The operator $(\Gamma^* \Gamma + \Upsilon^* \Upsilon)$ is in the form of a *skew-Toeplitz operator* that gives the name to this approach. See Foias et al. (1996) for a detailed exposition.

Optimal $\mathcal{H}_\infty$-Controller

The above steps have been implemented, and the final optimal controller expression has been obtained in a simplified form described below.

Let $\alpha_1, \ldots, \alpha_\ell \in \mathbb{C}_+$ be the zeros of $M_D(s)$, i.e., unstable poles of the plant (for simplicity of the exposition, they are assumed to be distinct). The sensitivity weight can be written as $W_1(s) = n W_1(s)/d W_1(s)$, for two coprime polynomials $n W_1$ and $d W_1$ with $\deg(n W_1) \le \deg(d W_1) =: n_1 \ge 1$. Define

$$E_\gamma(s) := \left(\frac{W_1(-s) W_1(s)}{\gamma^2} - 1 \right)$$

and let $\beta_1, \ldots, \beta_{2n_1}$ be the zeros of $E_\gamma(s)$, enumerated in such a way that $-\beta_{n_1+k} = \beta_k \in \overline{\mathbb{C}}_+$, for $k = 1, \ldots, n_1$. For notational convenience, assume that the zeros of E_γ are distinct for $\gamma = \gamma_{\mathrm{opt}}$.

Now define a rational function depending on $\gamma > 0$ and the weights W_1 and W_2,

$$F_\gamma(s) := \gamma \frac{d W_1(-s)}{n W_1(s)} G_\gamma(s)$$

where $G_\gamma \in \mathcal{H}_\infty$ is an outer function determined from the spectral factorization

$$G_\gamma(-s) G_\gamma(s)$$
$$= \left(1 + \frac{W_2(-s) W_2(s)}{W_1(-s) W_1(s)} - \frac{W_2(-s) W_2(s)}{\gamma^2} \right)^{-1}.$$

With the above definitions, under certain mild conditions (satisfied generically in most practical cases), the optimal controller can be expressed as

$$C_{\mathrm{opt}}(s) = \frac{E_\gamma(s) M_D(s) F_\gamma(s) L(s)}{1 + M_N(s) F_\gamma(s) L(s)} N_O^{-1}(s) \quad (4)$$

where $\gamma = \gamma_{\mathrm{opt}}$ and $L(s) = L_2(s)/L_1(s)$ with polynomials L_1 and L_2, of degree $n_1 + \ell - 1$, determined from the interpolation conditions:

$$L_1(\beta_k) + M_N(\beta_k) F_\gamma(\beta_k) L_2(\beta_k) = 0 \qquad k = 1, \ldots, n_1$$
$$L_1(\alpha_k) + M_N(\alpha_k) F_\gamma(\alpha_k) L_2(\alpha_k) = 0 \qquad k = 1, \ldots, \ell$$
$$L_2(-\beta_k) + M_N(\beta_k) F_\gamma(\beta_k) L_1(-\beta_k) = 0 \qquad k = 1, \ldots, n_1$$
$$L_2(-\alpha_k) + M_N(\alpha_k) F_\gamma(\alpha_k) L_1(-\alpha_k) = 0 \qquad k = 1, \ldots, \ell.$$

The above system of equations can be rewritten in the matrix form

$$\mathcal{R}_\gamma\, \Phi = 0 \tag{5}$$

where the $2(n_1 + \ell) \times 1$ vector Φ contains the coefficients of L_1 and L_2, and $\mathcal{R}_\gamma$ is a $2(n_1 + \ell) \times 2(n_1 + \ell)$ matrix which can be computed numerically when γ is fixed. The optimal performance level γ_{opt} is the largest γ which makes $\mathcal{R}_\gamma$ singular. The corresponding nonzero Φ gives $L(s)$, and hence, all the components of C_{opt} are computed.

Example 1 Consider the weighted sensitivity minimization for the plant (1) with the following first-order weights:

$$W_1(s) = \frac{1}{s}, \quad W_2(s) = k\, s \tag{6}$$

where $k > 0$ represents the relative importance of the multiplicative uncertainty with respect to the tracking performance under steplike reference inputs (see Doyle et al. 1992; Özbay 2000). With (6) the functions $E_\gamma(s)$ and $F_\gamma(s)$ are computed as

$$E_\gamma(s) = \frac{1 + \gamma^2 s^2}{-\gamma^2 s^2}, \quad F_\gamma(s) = \frac{-\gamma\, s}{k s^2 + k_\gamma s + 1},$$

where $\quad k_\gamma = \sqrt{2k - \dfrac{k^2}{\gamma^2}}. \tag{7}$

In this example $\ell = 1$ and $n_1 = 1$, with $\alpha_1 = 0.5$, $\beta_1 = j/\gamma$. For $k = 0.1$, the largest γ which makes $\mathcal{R}_\gamma$ singular is $\gamma_{\mathrm{opt}} = 7.452$, and the coefficients of the corresponding $L(s)$ are computed from the SVD of $\mathcal{R}_{\gamma_{\mathrm{opt}}}$,

$$L(s) = \frac{-0.0867 - 0.99623\, s}{-0.0867 + 0.99623\, s} = \frac{0.087 + s}{0.087 - s}.$$

Note that zeros of $E_\gamma(s) M_D(s)$ in $\overline{\mathbb{C}}_+$ appear as roots of the equation

$$1 + M_N(s) F_\gamma(s) L(s) = 0.$$

Hence, there are *internal* unstable pole-zero cancelations in the representation (4). An internally stable implementation of this controller is shown in Gumussoy (2011) using a realization similar to (3).

The above approach can also be extended to a class of infinite-dimensional plants with infinitely many poles in $\mathbb{C}_+$; see Gumussoy and Özbay (2004) for technical details.

Summary and Future Directions

This entry briefly summarized robust control problems involving linear time invariant infinite-dimensional plants with dynamic uncertainty models. Salient features of these robust control problems are captured by the mixed sensitivity minimization problem, for which a numerical computational procedure is outlined under the assumption that the weights are rational functions. Note that different types of plant models involving probabilistic, parametric, or structured (MIMO case) uncertainty are left out in this entry. Other robust control problems that are not discussed here include simultaneous stabilization (control of finitely many plant models by a single robust controller) and strong stabilization (robust control with the added restriction that the controller must be stable) of infinite-dimensional systems. Stable robust controller design techniques for different types of systems with time delays are illustrated in Özbay (2010) and Wakaiki et al. (2013); see also their references.

For practical implementation of infinite-dimensional robust controllers, it is important to find low-order approximations of stable irrational transfer functions with prescribed $\mathcal{H}_\infty$ error bound. There exist many different approximation techniques for various types of transfer functions, but there is still need for computationally efficient algorithms in this area. Another interesting topic along the same lines is direct computation of fixed-order $\mathcal{H}_\infty$ controllers for infinite-dimensional plants. In fact, computation of $\mathcal{H}_\infty$-optimal PID controllers is still a challenging

problem for infinite-dimensional plants, except for some time-delay systems satisfying certain simplifying structural assumptions. Advances in numerical optimization tools will play critical roles in the computation of low (or fixed)-order robust controller design for infinite-dimensional plants; see, e.g., Gumussoy and Michiels (2011) for recent results along this direction.

In the past, robust control of infinite-dimensional systems found applications in many different areas such as chemical processes, flexible structures, robotic systems, transportation systems, and aerospace. Robust control problems involving systems with time-varying and uncertain time delays appear in control of networks and control over networks. Ongoing research in the networked systems area include generalization of these problems to more complex and interconnected systems.

There are also many interesting robust control problems in biological systems, where typical underlying plant models are nonlinear and infinite dimensional. Some of these problems are solved under simplifying assumptions; it is expected that robust control theory will make significant contributions to this field by extensions of the existing results to more realistic plant and uncertainty models.

Cross-References

- ▶ Control of Linear Systems with Delays
- ▶ Flexible Robots
- ▶ H-Infinity Control
- ▶ Model Order Reduction: Techniques and Tools
- ▶ Networked Systems
- ▶ Optimal Control via Factorization and Model Matching
- ▶ Optimization Based Robust Control
- ▶ PID Control
- ▶ Robust Control in Gap Metric
- ▶ Spectral Factorization
- ▶ Stability and Performance of Complex Systems Affected by Parametric Uncertainty
- ▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Bibliography

Curtain R, Morris K (2009) Transfer functions of distributed parameter systems: a tutorial. Automatica 45:1101–1116

Curtain R, Zwart HJ (1995) An introduction to infinite-dimensional linear systems theory. Springer, New York

Desoer CA, Vidyasagar M (2009) Feedback systems: input-output properties. SIAM, Philadelphia

Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory. Macmillan, New York

Foias C, Özbay H, Tannenbaum A (1996) Robust control of infinite-dimensional systems: frequency domain methods. Lecture notes in control and information sciences, vol 209. Springer, London

Gu G, Khargonekar PP, Lee EB (1989) Approximation of infinite-dimensional systems. IEEE Trans Autom Control 34:610–618

Gumussoy S (2011) Coprime-inner/outer factorization of SISO time-delay systems and FIR structure of their optimal $\mathcal{H}_\infty$ controllers. Int J Robust Nonlinear Control 22:981–998

Gumussoy S, Michiels W (2011) Fixed-order H-infinity control for interconnected systems using delay differential algebraic equations. SIAM J Control Optim 49(5):2212–2238

Gumussoy S, Özbay H (2004) On the mixed sensitivity minimization for systems with infinitely many unstable modes. Syst Control Lett 53:211–216

Meinsma G, Mirkin L, Zhong Q-C (2002) Control of systems with I/O delay via reduction to a one-block problem. IEEE Trans Autom Control 47:1890–1895

Monje CA, Chen YQ, Vinagre BM, Xue D, Feliu V (2010) Fractional-order systems and controls, fundamentals and applications. Springer, London

Morris KA (2001) $\mathcal{H}_\infty$-output feedback of infinite-dimensional systems via approximation. Syst Control Lett 44:211–217

Özbay H (2000) Introduction to feedback control theory. CRC Press LLC, Boca Raton

Özbay H (2010) Stable $\mathcal{H}_\infty$ controller design for systems with time delays. In: Willems JC et al (eds) Perspectives in mathematical system theory, control, and signal processing. Lecture notes in control and information sciences, vol 398. Springer, Berlin/Heidelberg, pp 105–113

Sipahi R, Niculescu S-I, Abdallah CT, Michiels W, Gu K (2011) Stability and stabilization of systems with time delay. IEEE Control Syst Mag 31(1):38–65

Stein G (2003) Respect the unstable. IEEE Control Syst Mag 23(4):12–25

van Keulen B (1993) $\mathcal{H}_\infty$-control for distributed parameter systems: a state space approach. Birkhäuser, Boston

Wakaiki M, Yamamoto Y, Özbay H (2013) Stable controllers for robust stabilization of systems with infinitely many unstable poles. Syst Control Lett 62:511–516

R

Zhong QC (2006) Robust control of time-delay systems. Springer, London

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall, Upper Saddle River

Robust Fault Diagnosis and Control

Steven X. Ding
University of Duisburg-Essen, Duisburg, Germany

Abstract

Aiming at increasing system reliability and availability, integration of fault diagnosis into feedback control systems and integrated design of control and diagnosis receive considerable attention in research and industrial applications. In the framework of robust control, integrated diagnosis and control systems are designed to meet the demand for system robustness. The core of such systems is an observer that delivers needed information for a robust fault detection and feedback control.

Keywords

Observer-based fault diagnosis and control; Residual generation

Introduction

Advanced automatic control systems are marked by the high integration degree of digital electronics, intelligent sensors, and actuators. In parallel to this development, a new trend of integrating model-based fault detection and isolation (FDI) into the control systems can be observed (Blanke et al. 2006; Ding 2013; Gertler 1998; Isermann 2006; Patton et al. 2000), which is strongly driven by the enhanced needs for system reliability and availability.

A critical issue surrounding the integration of a diagnostic module into a feedback control loop is the interaction between the control and

diagnosis. Initiated by Nett et al. (1988), study on the integrated design of control and diagnosis has received much attention, both in the research and application domains. The original idea of the integrated design scheme proposed by Nett et al. (1988) is to manage the interactions between the control and diagnosis in an integrated manner (Ding 2009; Jacobson and Nett 1991).

Robustness is an essential performance for model-based control and diagnostic systems. In the control and diagnosis framework, robustness is often addressed in different context (Ding 2013) and thus calls for special attention in the integrated design of control and diagnostic systems. In their study on fault-tolerant controller architecture, Zhou and Ren (2001) have proposed to deal with the integrated design in the framework of the Youla parametrization of stabilization controllers (Zhou et al. 1996), which also builds the basis for achieving high robustness in an integrated control and diagnosis system. Below, we present the basic ideas and some representative schemes and methods for the integrated design of robust diagnosis and control systems.

Plant Model and Factorization Technique

Consider linear time invariant (LTI) systems given in the state space representation

$$\dot{x}(t) = Ax(t) + Bu(t) + E_d d(t) + E_f f(t)$$
$$y(t) = Cx(t) + Du(t) + F_d d(t) + F_f f(t)$$
$$z(t) = C_z x(t) + D_z u(t)$$

where $x \in \mathcal{R}^n, y \in \mathcal{R}^m, u \in \mathcal{R}^{k_u}$ stand for the plant state, output, and input vectors, respectively. $z \in \mathcal{R}^{k_z}$ is the controlled output vector. $d \in \mathcal{R}^{k_d}, f \in \mathcal{R}^{k_f}$ denote disturbance and fault vectors, respectively. $A, B, C, D, C_z, D_z, E_d, E_f, F_d, F_f$ are known matrices of appropriate dimensions.

A transfer matrix $G(s) = D + C(sI - A)^{-1}B$ with the minimal state space realization (A, B, C, D) can be factorized into

$$G(s) = \hat{M}^{-1}(s)\hat{N}(s)$$

$$\hat{M}(s) = I - C(sI - A_L)^{-1}L$$

$$\hat{N}(s) = D + C(sI - A_L)^{-1}B_L$$

$$A_L = A - LC, B_L = B - LD$$

where L is selected so that A_L is stable and can be interpreted as an observer gain matrix. This factorization is called left coprime (Zhou et al. 1996).

Parametrization of Stabilizing Controllers

Let

$$u(s) = K(s)y(s)$$

be an LTI feedback controller. By means of the well-known Youla parametrization (Zhou et al. 1996), all stabilizing controllers can be described and parametrized by

$$K(s) = \left(\hat{X}(s) - Q_c(s)\hat{N}(s)\right)^{-1} \cdot$$
$$\left(\hat{Y}(s) - Q_c(s)\hat{M}(s)\right)$$

$$\hat{X}(s) = I - F(sI - A_L)^{-1}B_L$$

$$\hat{Y}(s) = F(sI - A_L)^{-1}L$$

where $Q_c(s)$ is a stable parameter matrix, and F is selected so that $A_F = A + BF$ is stable and can be interpreted as a state feedback gain matrix.

Parametrizations of Residual Generators

Given the system under consideration, an LTI residual generator is a dynamic system with $u(t), y(t)$ as its inputs and $r(t)$ as output which satisfies, for $d(t) = 0, f(t) = 0$,

$$\forall x(0), u(t), \lim_{t \to \infty} r(t) = 0.$$

Residual generation is the first step for a successful fault diagnosis. The generated residual vector is an indicator for the occurrence of a fault. It is well known that all LTI residual generators can be parametrized by

$$r(s) = R(s)\left(\hat{M}(s)y(s) - \hat{N}(s)u(s)\right)$$

where $R(s)$ is a stable parameter matrix and called post-filter (Ding 2013).

Integration of Controller and Residual Generator into a Control Loop

It is remarkable that both the feedback controllers and residual generators can be parametrized based on the left coprime factorization of the plant model. This is the basis for an integration of diagnosis and control into a feedback control system. In Ding et al. (2010), it is demonstrated that the abovementioned Youla parametrization form is in fact an observer-based feedback controller, which can be expressed by

$$u(s) = F\hat{x}(s) + Q_c(s)\left(\hat{N}(s)u(s) - \hat{M}(s)y(s)\right)$$

where $\hat{x}(s)$ is a state estimate delivered by a full-order state observer (Anderson 1998; Zhou et al. 1996). Moreover, the residual generator can also be written as

$$r(s) = R(s)r_o(s), r_o(s) = y(s) - \hat{y}(s)$$

with $\hat{y}(s)$ being the output estimate delivered by an observer (Ding 2013). As a result, a stabilization feedback controller and residual generator can be integrated into a dynamic system of the following form:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + Lr_o(t)$$
$$= A_L\hat{x}(t) + B_Lu(t) + Ly(t)$$

$$r_o(t) = y(t) - \hat{y}(t), \hat{y}(t) = C\hat{x}(t) + Du(t)$$

$$\begin{bmatrix} u(s) \\ r(s) \end{bmatrix} = \begin{bmatrix} F\hat{x}(s) \\ 0 \end{bmatrix} + \begin{bmatrix} -Q_c(s) \\ R(s) \end{bmatrix} r_o(s).$$

The core of the above control and diagnostic system is a state observer that delivers a state estimation $\hat{x}(t)$ and the primary residual vector $r_o(t)$. The design parameters of this integrated control and diagnosis system are L, F; the observer and state feedback control gain matrices, as well as $Q_c(s), R(s)$.

Robustness of Diagnostic and Control Systems

While in the robust control framework, the controller design is typically formulated as minimizing a system norm of the transfer function matrix from the disturbance vector d to the control output z (Zhou et al. 1996), the design objective of a robust fault detection system consists in an optimal trade-off between the robustness against d and the sensitivity to the fault vector f. Considering that

$$r(s) = R(s)\,(y(s) - \hat{y}(s))$$

$$= R(s)\left(\hat{N}_d(s)d(s) + \hat{N}_f(s)f(s)\right)$$

$$\hat{N}_d(s) = F_d + C\,(sI - A_L)^{-1}\,(E_d - LF_d)$$

$$\hat{N}_f(s) = F_f + C\,(sI - A_L)^{-1}\,(E_f - LF_f)$$

Ding (2013), the design objective can be formulated as

$$\sup_{R(s)} \frac{\left\| R(s)\hat{N}_f(s) \right\|_{\text{index}}}{\left\| R(s)\hat{N}_d(s) \right\|}$$

or in a suboptimum form as finding $R(s)$ so that for some given $\alpha > 0, \beta > 0$

$$\left\| R(s)\hat{N}_d(s) \right\| \le \alpha, \left\| R(s)\hat{N}_f(s) \right\|_{\text{index}} > \beta.$$

Similar to the robust controller design, a (system) norm like $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm, denoted by $\|\cdot\|$, is applied for the evaluation of the influence of the disturbances. Differently, the evaluation of the sensitivity to the fault vector, expressed by $R(s)\hat{N}_f(s)$, can be realized using either a system norm or the so-called $\mathcal{H}_-$ index, denoted by $\|\cdot\|_-$, which indicates the minimum influence of f on r (Ding 2013).

In order to detect the fault occurrence reliably and successfully, a decision-making procedure is needed. It consists of a further evaluation of the residual signal and a detection logic. Typically, a signal norm of r, e.g., $\mathcal{L}_2$ norm, and a simple detection logic like

$$\begin{cases} \|r\| > J_{th} \Longrightarrow \text{Alarm for fault} \\ \|r\| \le J_{th} \Longrightarrow \text{Fault-free} \end{cases}$$

are adopted for this purpose, where J_{th} is a further design parameter and called threshold (Ding 2013). The threshold setting depends on the dynamics of r, its norm-based evaluation, and has significant influence on the fault detection performance. For the purpose of reducing false alarms, the threshold is often set as

$$J_{th} = \sup_{f=0, \|d\| \le d_d} \|r\|$$

$$= \sup_{f=0, \|d\| \le d_d} \left\| R(s)\hat{N}_d(s)d(s) \right\|.$$

That is, the threshold is set to be the maximum value of the influence of the disturbances on the residual signal in the fault-free case. Thus, different designs of the residual generator will result in different threshold settings. In this context, an optimal design of a fault diagnosis system is understood as an integrated design of the residual generator, the evaluation function, and the threshold (Ding 2013).

An Integrated Design Scheme for Robust Diagnosis and Control

Assume that the system under our consideration satisfies the following conditions:

- $\|d\|_2 \le \delta_d$.
- (A, B) is stabilizable and (C, A) is detectable.
- $D = 0$.
- $D_z^T D_z > 0$ and $F_d F_d^T > 0$.
- $\begin{bmatrix} A - j\omega I & B \\ C_z & D \end{bmatrix}$ has full column rank for all ω.
- $\begin{bmatrix} A - j\omega I & E_d \\ C & F_d \end{bmatrix}$ has full row rank for all ω.

Then, the following observer and state feedback gain matrices

$$L^* = \left(E_d F_d^T + Y C^T\right)\left(F_d F_d^T\right)^{-1}$$
$$F^* = -\left(D_z^T D_z\right)^{-1}\left(B^T X + D_z^T C_z\right)$$

as well as

$$Q_c^*(s) = 0, \; R^*(s) = \left(F_d F_d^T\right)^{-1/2}$$

result in an optimal integrated design of the robust diagnostic and control system with

- $\mathcal{H}_2$ optimal control performance
- Maximal fault detectability and the optimal threshold setting

$$J_{th} = \sup_{f=0, \|d\| \le \delta_d} \|r\|_2 = \delta_d$$

where $Y \ge 0$, $X \ge 0$ are respectively the solution of the following two Riccati equations:

$$AY + YA^T + E_d E_d^T - \left(E_d F_d^T + Y C^T\right) \cdot$$
$$\left(F_d F_d^T\right)^{-1}\left(E_d F_d^T + Y C^T\right)^T = 0$$
$$A^T X + XA + C_z^T C_z - \left(C_z^T D_z + XB\right) \cdot$$
$$\left(D_z^T D_z\right)^{-1}\left(C_z^T D_z + XB\right)^T = 0.$$

That L^*, F^* lead to minimizing the $\mathcal{H}_2$ norm of the transfer matrix from d to z is a well-known result (Zhou et al. 1996). The optimal fault detection performance can be understood from two different viewpoints:

- Optimum in the sense of

$$\forall \omega, \quad \sup_{R(s)} \frac{\sigma_i\left(R(j\omega)\hat{N}_f(j\omega)\right)}{\left\| R(s)\hat{N}_d(s)\right\|_\infty}$$
$$= \sigma_i\left(\left(F_d F_d^T\right)^{-1/2} \hat{N}_f^*(j\omega)\right)$$

where $\sigma_i\left(R(j\omega)\hat{N}_f(j\omega)\right)$ is the i-th singular value of matrix $R(j\omega)\hat{N}_f(j\omega)$, $i = 1, \cdots, k_f$, $\hat{N}_f^*(s) = \hat{N}_f(s)|_{L=L^*}$ (Ding 2013).

- A fault that can be detected by any LTI detection system will also be detected using the detection system with the above parameter and threshold setting. Thus, this detection system provides the maximal fault detectability (Ding 2013).

It is worth remarking that:

- The assumptions mentioned above are standard in the $\mathcal{H}_2$ optimal control (Zhou et al. 1996).
- The optimization problem

$$\forall \omega, \sup_{R(s)} \frac{\sigma_i\left(R(j\omega)\hat{N}_f(j\omega)\right)}{\left\| R(s)\hat{N}_d(s)\right\|_\infty}$$

is a more general form of the so-called $\mathcal{H}_-/\mathcal{H}_\infty$ or $\mathcal{H}_\infty/\mathcal{H}_\infty$ optimization of observer-based fault detection systems, and thus, its solution is called unified solution (Ding 2013).

- The solution given above is a state space realization of the robust fault detection problems, which is e.g., described by Ding (2013) in Theorem 7.16.
- This integrated design scheme can also be applied to discrete-time and stochastic systems (Ding 2013).

Summary and Future Directions

Increasing reliability and availability of advanced automatic control systems is of considerable practical interests. Integration of fault diagnosis into feedback control systems and integrated design of robust control and diagnosis are useful solutions for real-time applications (Ding 2009). They can also be integrated into a fault-tolerant control system (Blanke et al. 2006; Zhou and Ren 2001). A further potential application field is fault diagnosis in feedback control loops using embedded residual signals (Ding et al. 2010).

From the viewpoint of research, integrated design of robust control and diagnosis in nonlinear and time-varying dynamic systems

are challenging issues. The $\mathcal{L}_2$-gain technique for nonlinear control (Van der Schaft 2000) and the fault detection scheme proposed by Li and Zhou (2009) are promising and useful results for the future investigations in this area.

Cross-References

▶ Fault Detection and Diagnosis
▶ Fault-Tolerant Control
▶ Robust $\mathcal{H}_2$ Performance in Feedback Control

Bibliography

Anderson BDO (1998) From Youla-Kucera to identification, adaptive and nonlinear control. Automatica 34:1485–1506
Blanke M, Kinnaert M, Lunze J, Staroswiecki M (2006) Diagnosis and fault-tolerant control, 2nd edn. Springer, Berlin/New York
Ding SX (2009) Integrated design of feedback controllers and fault detectors. Ann Rev Control 33:124–135
Ding SX (2013) Model-based fault diagnosis techniques – design schemes, algorithms and tools, 2nd edn. Springer, London
Ding SX, Yang G, Zhang P, Ding EL, Jeinsch T, Weinhold N, Schulalbers M (2010) Feedback control structures, embedded residual signals and feedback control schemes with an integrated residual access. IEEE Trans Control Syst Technol 18:352–367
Gertler JJ (1998) Fault detection and diagnosis in engineering systems. Marcel Dekker, New York
Isermann R (2006) Fault diagnosis systems. Springer, Berlin Heidelberg
Jacobson CA, Nett CN (1991) An integrated approach to controls and diagnostics using the four parameter controller. IEEE Control Syst 11:22–28
Li X, Zhou K (2009) A time domain approach to robust fault detection of linear time-varying systems. Automatica 45:94–102
Nett CN, Jacobson CA, Miller AT (1988) An integrated approach to controls and diagnostics. In: Proceedings of ACC, Atlanta, Georgia, pp 824–835
Patton RJ, Frank PM, Clark RN (eds) (2000) Issues of fault diagnosis for dynamic systems. Springer, London
van der Schaft A (2000) L2 – gain and passivity techniques in nonlinear control. Springer, London
Zhou K, Ren Z (2001) A new controller architecture for high performance, robust, and fault-tolerant control. IEEE Trans Autom Control 46:1613–1618
Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall, Upper Saddle River

Robust $\mathcal{H}_2$ Performance in Feedback Control

Fernando Paganini
Universidad ORT Uruguay, Montevideo, Uruguay

Abstract

This entry discusses an important compromise in feedback design: reconciling the superior performance characteristics of the $\mathcal{H}_2$ optimization criterion, with robustness requirements expressed through induced norms such as $\mathcal{H}_\infty$. The fact that both criteria have frequency-domain characterizations and involve similar state-space machinery motivated many researchers to seek an adequate combination. We review here robust $\mathcal{H}_2$ analysis methods based on convex optimization developed in the 1990s and comment on their implications for controller synthesis.

Keywords

Linear matrix inequalities; Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control; Robustness analysis; Structured uncertainty

Introduction

Can mathematics help us deal with the inevitable theory-practice gap? Should we be optimistic and assume that discrepancies between models and nature are random and neutral towards our actions or be pessimistic and design for the worst such discrepancies? Feedback control theory has struggled with these questions, perhaps more so than other fields.

During the surge of optimal control in the 1960s, optimism carried the day. A prominent example is the LQG ($\mathcal{H}_2$) regulator, which minimizes the effect of random disturbances and has an elegant state-space solution; in comparison, the frequency-domain designs of classical control appeared primitive and conservative. But the

pessimists struck back in the late 1970s, showing things could go very wrong (unstable) with LQG, when a parameter variation was introduced in the plant model. This ushered in the robust control era of the 1980s, with its worst-case analysis of stability over deterministic sets of plants, leading to other design metrics such as $\mathcal{H}_\infty$ control. In this mentality, exogenous disturbances were also treated as an adversary to be protected against in the worst case, perhaps an excess of pessimism.

The robust $\mathcal{H}_2$ problem incarnates the search for a middle ground, where stability is treated with the conservatism it deserves, but performance is optimized for a more neutral noise. This entry summarizes efforts made around the 1990s to seek this compromise.
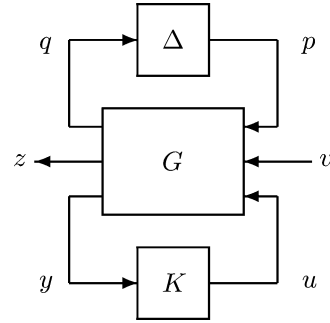
$\mathcal{H}_2$ Optimal Control

In the feedback diagram of Fig. 1, signals are vector valued, and we focus on continuous time. G is a linear system with a given state-space representation. Initially omit the upper loop (set $\Delta = 0$). The LQG regulator is the controller K that internally stabilizes the feedback loop and minimizes the variance of the error variable z, assuming the input v is white Gaussian noise.

For an alternative description, denote by $\hat{T}_{zv}(s)$ the closed-loop transfer function from v to z; we wish to design K such that $\hat{T}_{zv}(s)$ is analytic in $Re(s) \geq 0$ and has minimum $\mathcal{H}_2$ norm, defined by

$$\|\hat{T}_{zv}\|_{\mathcal{H}_2} = \left(\int_{-\infty}^{\infty} \mathbf{Tr}(\hat{T}_{zv}(j\omega)^* \hat{T}_{zv}(j\omega)) \frac{d\omega}{2\pi} \right)^{\frac{1}{2}};$$
$$(1)$$

here **Tr** denotes matrix trace and * denotes conjugate transpose. The equivalence between this $\mathcal{H}_2$-optimal control and LQG follows from classical filtering, modeling v as uncorrelated components of unit power spectral density over all frequency. By adding a filter in the input of G, noise of known, colored spectrum can be accommodated as well.

A different motivation, in the case of scalar v, is to observe that $\|\hat{T}_{zv}\|_{\mathcal{H}_2}^2$ is the energy ($\mathcal{L}_2$-norm square) of the system impulse response.

Robust $\mathcal{H}_2$ Performance in Feedback Control, Fig. 1
Feedback control and model uncertainty

Thus it measures the transient error in response to known inputs or initial conditions which may be generated by an impulse.

The $\mathcal{H}_2$ (LQG) optimal feedback has an elegant solution, computable in state-space through two algebraic Riccati equations (AREs). Its quick popularity was, however, hampered due to its lack of *stability margins*: a small error in model parameters can make the closed-loop unstable (Doyle 1978). This motivated methods to explicitly address such modeling errors.

Model Uncertainty and Robustness

Suppose some parameter in the model of G is uncertain, $\alpha = \alpha_0 + \kappa\delta$, $\delta \in [-1, 1]$; often, the normalized variation δ can be "pulled out" into the *uncertainty block* Δ of Fig. 1. The same technique can account for unmodeled linear time-invariant (LTI) dynamics, e.g., high frequency effects: they can be "covered" by a normalized transfer function $\hat{\Delta}(j\omega)$ and frequency weights that connect it to G. Even further, a nonlinear or time-varying (NL,TV) modeling error can be represented through an *operator* Δ in signal space. The references contain details on this modeling technique.

To analyze the effect of such errors, suppose K has been chosen to stabilize G and M is the resulting closed-loop system, with state-space representation

$$\dot{x} = Ax + B_p p + B_v v, \qquad (2)$$
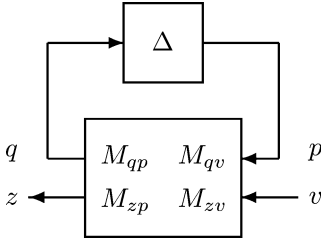
Robust $\mathcal{H}_2$ Performance in Feedback Control, Fig. 2
Robustness analysis setup

$$q = C_q x,$$
$$z = C_z x.$$

A is an $n \times n$ stable (Hurwitz) matrix, and for simplicity there are no feed-through terms. Figure 2, represents the interconnection of M with the uncertainty.

To quantify the size of uncertainty, it is convenient to use an *induced norm* (gain) in signal space and constrain Δ to the normalized ball $\{\|\Delta\| \leq 1\}$. If the subsystem M_{qp} in feedback with Δ satisfies itself the induced norm constraint $\|M_{qp}\| < 1$, the small gain theorem implies *robust stability* over the entire ball. Focusing for the rest of this article on the $\mathcal{L}_2$ signal space (square-integrable functions), the latter induced norm is equivalent to the $\mathcal{H}_\infty$ norm of the transfer function:

$$\|\hat{M}_{qp}(s)\|_{\mathcal{H}_\infty} := \operatorname*{ess\,sup}_{\omega \in \mathbb{R}} \bar{\sigma}(\hat{M}_{qp}(j\omega)),$$

where $\bar{\sigma}(\cdot)$ denotes matrix maximum singular value.

This motivates $\mathcal{H}_\infty$-optimal control: design K to minimize the above quantity with internal stability. This problem also admits state-space solutions based on AREs and thus is a valid competing paradigm to $\mathcal{H}_2$.

To accommodate multiple sources of uncertainty within Fig. 1, we can use a *block diagonal* structure:

$$\Delta = \operatorname{diag}[\Delta_1, \ldots, \Delta_d]. \tag{3}$$

Here, different uncertainty blocks (parametric, LTI, LTV, or NL) enter in separate "channels"; $\mathbf{B_\Delta}$ denotes the unit ball of operators with the prescribed structure. For stability studies, *causality* of the operator is required.

Robust stability under structured uncertainty is a rich topic: we refer to the article on the *structured singular value* (μ) in this encyclopedia. We invoke here robustness conditions based on the set Λ of positive definite matrix *scalings* or *multipliers* of the form:

$$\Lambda = \operatorname{diag}[\lambda_1 I, \ldots, \lambda_d I], \tag{4}$$

with submatrices of the same dimensions as the blocks in (3), thus commuting with a matrix Δ of that structure.

Consider the frequency family of matrix inequalities

$$\hat{M}_{qp}^*(j\omega)\Lambda(\omega)\hat{M}_{qp}(j\omega) - \Lambda(\omega) < 0 \quad \forall \omega;$$
$$\Lambda(\omega) \in \Lambda. \tag{5}$$

At each ω, this is a linear matrix inequality (LMI); testing its feasibility is a convex, tractable problem. A solution implies the *scaled-small gain condition*

$$\bar{\sigma}\left(\Lambda(\omega)^{\frac{1}{2}}\hat{M}_{qp}(j\omega)\Lambda^{-\frac{1}{2}}(\omega)\right) < 1;$$

this "μ upper bound" implies robust stability when uncertainty is LTI, through commuting $\Lambda(\omega)$ with $\hat{\Delta}(j\omega)$.

If uncertainty is NLTV, (5) must be strengthened to enforce $\Lambda(\omega) \equiv \Lambda$, constant in frequency. This condition turns out to be both necessary and sufficient for robust stability. Here the LMIs would be coupled in frequency; however, the Kalman-Yakubovich-Popov lemma reduces them to an equivalent LMI in terms of the state-space matrices in (2), with variables $\Lambda \in \mathbf{\Lambda}$ and an $n \times n$ matrix $P > 0$:

$$\begin{bmatrix} A^*P + PA + C_q^*\Lambda C_q & PB_p \\ B_p^*P & -\Lambda \end{bmatrix} < 0. \tag{6}$$

What about performance? The mapping $T_{zv}(\Delta)$ between the disturbance v and the error z now depends on the uncertainty. The default procedure in robust control has been to measure performance with the same induced norm, evaluating $\|T_{zv}(\Delta)\|_{\mathcal{L}_2 \to \mathcal{L}_2}$ in the worst-case over $\Delta \in \mathbf{B_\Delta}$. This can be computed with similar complexity to establishing robust stability. It amounts, however, to treating noise with the same worst-case mentality as stability, a questionable choice. For instance, in LTI systems the worst-case signals are sinusoids at the worst frequency and spatial direction; while one should protect against such signals arising in the Δ-loop due to instability, it is not natural to expect them as external disturbances, which are usually of broad spectrum.

Robust $\mathcal{H}_2$ Performance Analysis

In the absence of uncertainty, the $\mathcal{H}_2$ norm of the nominal mapping $T_{zv}(0) = M_{zv}$ provides a natural performance criterion, measuring the response to flat-spectrum disturbances or the transient response. When uncertainty is present, it motivates a worst-case analysis of stability; a natural combination is to impose *robust $\mathcal{H}_2$ performance*: evaluating the worst-case $\mathcal{H}_2$ norm of $T_{zv}(\Delta)$ over the uncertainty class $\mathbf{B_\Delta}$. We will highlight some methods based on semidefinite programming to perform such evaluations; for further details and comparisons, we refer to Paganini and Feron (1999).

A Frequency Domain Robust Performance Criterion

Consider the following optimization:

$$\mathbf{J_f} := \inf \int_{-\infty}^{\infty} \mathbf{Tr}(Y(\omega))\frac{d\omega}{2\pi}, \text{ subject to}$$

$$\hat{M}(j\omega)^* \begin{bmatrix} \Lambda(\omega) & 0 \\ 0 & I \end{bmatrix} \hat{M}(j\omega) - \begin{bmatrix} \Lambda(\omega) & 0 \\ 0 & Y(\omega) \end{bmatrix} < 0 \tag{7}$$

for each ω, and $\Lambda(\omega) \in \Lambda$.

Here $\hat{M}$ is the transfer function in Fig. 2; a submatrix of the above includes (5), implying

robust stability under structured LTI uncertainty. Furthermore, we have the robust $\mathcal{H}_2$ performance bound (Paganini 1999):

$$\sup_{\Delta \in \mathbf{B_\Delta^{LTI}}} \|T_{zv}(\Delta)\|_{\mathcal{H}_2}^2 \le \mathbf{J_f}. \tag{8}$$

We sketch the argument based on the Fourier transforms $\hat{p}(j\omega)$, etc., for signals in Fig. 2. Applying the quadratic form in (7) to the joint vector of $\hat{p}$ and $\hat{v}$ gives

$$\sum_{i=1}^{d} \lambda_i(\omega)|\hat{q}_i|^2 + |\hat{z}|^2 \le \sum_{i=1}^{d} \lambda_i(\omega)|\hat{p}_i|^2 + \hat{v}^* Y(\omega)\hat{v}.$$

The subvectors $\hat{p}_i$, $\hat{q}_i$ correspond to uncertainty blocks, $\hat{p}_i = \hat{\Delta}_i(j\omega)\hat{q}_i$; since $\bar{\sigma}(\hat{\Delta}_i(j\omega)) \le 1$, $|\hat{q}_i| \ge |\hat{p}_i|$. Also $\lambda_i(\omega) > 0$, so these terms can be simplified, leading to

$$|\hat{T}_{zv}(j\omega)\hat{v}|^2 = |\hat{z}|^2 \le \hat{v}^* Y(\omega)\hat{v}.$$

This means $\hat{T}_{zv}(j\omega)^* \hat{T}_{zv}(j\omega) \le Y(\omega)$ for every Δ, and therefore the $\mathcal{H}_2$ norm bound

$$\|T_{zv}(\Delta)\|_{\mathcal{H}_2}^2 \le \int_{-\infty}^{\infty} \mathbf{Tr}(Y(\omega))\frac{d\omega}{2\pi}$$

holds, from which (8) follows.

The computation involved in (7) at each frequency is a semidefinite program (SDP): minimizing the linear cost $\mathbf{Tr}(Y(\omega))$ subject to an LMI constraint, a tractable problem. Adding a frequency sweep, we have a practical method to bound the desired robust performance.

The inequality (8) is in general strict. Beyond the usual conservatism of convex bounds for μ, when noise is of dimension m, a conservatism of up to this order may appear; an improvement to address this issue with augmented SDPs is given in Sznaier et al. (2002). Finally, causality of the uncertainty is not imposed in the frequency-domain criterion.

As in the study of robust stability, we wish to extend the analysis to NLTV uncertainty blocks. Now the mapping $T_{zv}(\Delta)$ can no longer be represented by a transfer function, so what is the "$\mathcal{H}_2$" cost? We return to our motivation for this

performance notion: to measure the effect of disturbances of flat spectrum.

In Paganini (1999), the flat-spectrum property is imposed as a deterministic constraint on the input disturbances. For the scalar v case, define $W_{\eta,B} \subset \mathcal{L}_2$ by the family of *integral quadratic constraints*:

$$\int_{-\beta}^{\beta} |v(j\omega)|^2 \frac{d\omega}{2\pi} \begin{cases} \leq \frac{\beta}{\pi} + \eta & \forall \beta; \\ \geq \frac{\beta}{\pi} - \eta, & \beta \in [0, B]. \end{cases} \quad (9)$$

This imposes that the cumulative spectrum is approximately linear (to a tolerance $\eta > 0$), up to bandwidth B, and has sublinear growth beyond that. Extensions to vector-valued signals are also given. For a stable LTI system T_{zv}, it is not difficult to verify that

$$\|T_{zv}\|_{\mathcal{H}_2}^2 = \lim_{\substack{\eta \to 0 \\ B \to \infty}} \sup_{v \in W_{\eta,B}} \|T_{zv} v\|_2^2,$$

but the right-hand side applies to NLTV systems as well. The following result can be established in the latter case:

$$\lim_{\substack{\eta \to 0 \\ B \to \infty}} \sup_{v \in W_{\eta,B}, \Delta \in \mathbf{B}_\Delta^{\mathbf{NLTV}}} \|T_{zv}(\Delta)v\|_2^2 = \mathbf{J}_\mathbf{f}',$$

where the right-hand side is the variant of (7) with the restriction that $\Lambda(\omega) \equiv \Lambda$, constant in frequency. In this case the characterization is *exact*, with equality above. This follows from a duality argument in function space, where $Y(\omega)$ appears as the multiplier for the constraint in (9). While coupled in frequency, $\mathbf{J}_\mathbf{f}'$ is again equivalent to a finite-dimensional SDP in state space.

Let us review, instead, a different state-space method, motivated by alternate definitions of the $\mathcal{H}_2$ cost.

A State-Space Criterion Invoking Causality

Consider the semidefinite program

$$\mathbf{J}_\mathbf{s} := \inf \, \mathbf{Tr}(B_v^* P B_v) \text{ subject to } P > 0, \Lambda \in \Lambda,$$

$$\begin{bmatrix} A^* P + PA + C_q^* \Lambda C_q + C_z^* C_z & PB_p \\ B_p^* P & -\Lambda \end{bmatrix} < 0. \quad (10)$$

The LMI above is very similar to (6); indeed it provides a robust stability certificate and in addition a bound on a generalized $\mathcal{H}_2$ cost, for arbitrary (NLTV) causal uncertainty blocks. Again, we sketch the argument.

For stability, consider the system of Fig. 2 with $v \equiv 0$, initial condition $x(0) = x_0$. Define the storage function $V(x) = x^* P x$; differentiating it under (2) and applying the LMI (10) to the joint vector of $x(t)$, $p(t)$ yield

$$\dot{V} + |z|^2 \leq -q^* \Lambda q + p^* \Lambda p = \sum_{i=1}^{d} \lambda_i (|p_i|^2 - |q_i|^2).$$

Integrating the above over $(0, t)$, the sum on the right becomes nonpositive because $\lambda_i > 0$ and the operator $\Delta_i : q_i \to p_i$ is causal and contractive. This leads to

$$V(x(t)) + \int_0^t |z(\tau)|^2 d\tau \leq V(x_0), \quad (11)$$

which implies Lyapunov stability; the bound can be sharpened to prove asymptotic stability. Also, letting $t \to \infty$ yields the energy bound $\|z\|_2^2 \leq V(x_0)$.

Suppose now that x_0 is generated by applying to the (causal) system at rest, an impulse $v(t) = \delta(t)$, assumed scalar. The result is $x(0+) = B_v$, so $V(x_0) = B_v^* P B_v$; the impulse response energy of $T_{zv}(\Delta)$ is thus bounded. Minimizing over P, Λ leads to the robust $\mathcal{H}_2$ performance bound

$$\sup_{\Delta \in \mathbf{B}_\Delta^{\mathbf{NLTV}}} \|T_{zv}(\Delta)\delta(t)\|_2^2 \leq \mathbf{J}_\mathbf{s},$$

where the $\mathcal{H}_2$ cost is generalized as the impulse response energy. An extension to multiple impulse channels is available. This kind of result was first obtained by Stoorvogel (1993) for unstructured uncertainty.

An alternate notion of $\mathcal{H}_2$ cost for NLTV systems, also considered in Stoorvogel (1993), is the average output variance when the input

is random white noise. This is formalized by replacing (2) with a stochastic differential equation (e.g., Oksendal 1985) and extending the bound (11) using Ito calculus; for details see Paganini and Feron (1999). The following robust $\mathcal{H}_2$ performance bound is obtained:

$$\limsup_{\tau \to \infty} \frac{1}{\tau} \int_0^\tau E|z(t)|^2 dt. \leq \mathbf{J_s} \quad \forall \Delta \in \mathbf{B_\Delta^{NLTV}}.$$

What if the uncertainty is time invariant? Incorporating frequency-dependent scalings, with causality, into the state-space approach must be done approximately, generating $\hat{\Lambda}(j\omega)$ through the span of a predefined finite basis of causal, rational transfer functions. Searching over this basis for a bound on the impulse response energy can be pursued with state-space SDPs, now of a size increasing with the basis dimensionality. We refer to Feron (1997) for details.

Robust $\mathcal{H}_2$ Synthesis

Prior sections have focused on the robustness *analysis* of a closed-loop system M, obtained from G after designing a nominally stabilizing controller. Can we *synthesize K* with robust $\mathcal{H}_2$ performance as an objective? We overview some contributions to this question.

Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ Control

Let us discuss first the more modest objective of optimizing *nominal* $\mathcal{H}_2$ performance while guaranteeing robust stability. If the uncertainty block Δ in Fig. 2 is unstructured, the problem is equivalent to

Minimize $\|\hat{M}_{zv}\|_{\mathcal{H}_2}$, subject to $\|\hat{M}_{qp}\|_{\mathcal{H}_\infty} < 1$.

Using a Youla parameterization of stabilizing controllers, $\hat{M}(s)$ depends affinely on a stable parameter $\hat{Q}(s)$; this makes the optimization over $\hat{Q}$ convex. However it has been shown to give infinite-dimensional solutions that must be approximated by suitable truncations; see Sznaier et al. (2000) and references therein.

To better exploit the state-space structure common to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis, Bernstein and Haddad (1989) proposed a simplification: minimize an *auxiliary cost* that upper bounds the $\mathcal{H}_2$ norm while imposing the $\mathcal{H}_\infty$ constraint, through a common storage function. This cost is optimized by controllers of the order of the plant, characterized in terms of coupled AREs; later on Khargonekar and Rotea (1991) recast this problem using convex optimization. Also Zhou et al. (1994) and Doyle et al. (1994) studied the dual (transpose) structure.

The latter version is in fact directly related to the analysis condition (10), with a fixed $\Lambda = \lambda I$. A matrix P satisfying this condition imposes the $\mathcal{H}_\infty$ norm restriction and upper bounds the nominal $\mathcal{H}_2$ cost. This idea of imposing multiple objectives through a common storage function has more general applicability: Scherer et al. (1997) showed that all such problems admit tractable synthesis based on LMIs, with solutions of the same order as the plant.

Synthesis for Robust Performance

We have seen that rather than just an upper bound on nominal performance, (10) ensures the more stringent robust $\mathcal{H}_2$ performance requirement; therefore it becomes the basis of a robust $\mathcal{H}_2$ synthesis technique. In Stoorvogel (1993) this method is laid out for unstructured uncertainty: search linearly over the scalar λ and solve the auxiliary cost synthesis problem for each λ.

What about structured uncertainty? We run here into a general difficulty of such synthesis questions, even for robust stability alone. In that case, seeking simultaneously a controller K and a scaling Λ so that conditions (5) or (6) are satisfied by the resulting M is not a computationally friendly problem. In the absence of a general solution method, iterating between an $\mathcal{H}_\infty$ design of K for fixed Λ and the analysis conditions to find Λ is commonly used for design.

Things can be no easier for robust $\mathcal{H}_2$ performance, but the iterative procedure does generalize to the conditions in (10): for fixed K, the SDP will return structured Λ's, which can then be fixed for a multiobjective synthesis step based on the "auxiliary cost" in (10) as discussed above. If constant Λ are used (designing for NLTV

uncertainty), all controllers obtained are of the order of the plant.

If uncertainty is LTI, an alternative is to carry out the analysis step in the frequency domain, finding a $\Lambda(\omega)$, $Y(\omega)$ through (7). In the corresponding situation for μ-synthesis, where only $\Lambda(\omega)$ is found, a step of fitting and spectral factorization is needed to approximate such scalings through a rational weights, which are then incorporated into $\mathcal{H}_\infty$ synthesis. A similar frequency weight in the performance channel can approximate the effect of $Y(\omega)$, thus relying on weighted $\mathcal{H}_\infty$ synthesis to pursue the $\mathcal{H}_2$ performance objective. Of course, the order of the resulting controllers is increased.

Summary and Future Directions

The tradeoff between performance and robustness is essential to feedback control. In the case of linear multivariable design, it motivated a compromise between $\mathcal{H}_2$ performance and $\mathcal{H}_\infty$-type robustness, pursued with the state-space and frequency-domain tools common to these metrics. We have highlighted robust $\mathcal{H}_2$ analysis conditions obtained in the 1990s based on semidefinite programming, which provided the greatest flexibility to integrate the aforementioned tools and different points of view (worst-case, average case) present in this problem. As in other situations, the robust synthesis question has proven more difficult: design cannot be "automated" to the degree that was once envisioned.

The passage of time makes issues that once attracted strong attention look narrow in scope, so it is not natural to indicate directions that directly follow on this work. Perhaps the best legacy that the robust $\mathcal{H}_2$ generation can take to other problems is the willingness to integrate various disciplines (dynamics, operator theory, stochastics, optimization) to face the demands of applied mathematical research.

Cross-References

▶ H-Infinity Control
▶ KYP Lemma and Generalizations/Applications
▶ Linear Quadratic Optimal Control
▶ LMI Approach to Robust Control
▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Recommended Reading

LQG control is covered in many textbooks, e.g., Anderson and Moore (1990). A standard text for robust control with an $\mathcal{H}_\infty$ perspective, including structured singular values, the Youla parameterization, and the Riccati equation solution for $\mathcal{H}_\infty$ synthesis, is Zhou et al. (1996); see also Sánchez-Peña and Sznaier (1998) with application examples. The textbook of Dullerud and Paganini (2000) incorporates the more recent developments based on LMIs; see Boyd and Vandenberghe (2004) for background on semidefinite programming.

Bibliography

Anderson B, Moore JB (1990) Optimal control: linear quadratic methods. Prentice Hall, Englewood Cliffs

Bernstein DS, Haddad WH (1989) LQG control with an $\mathcal{H}_\infty$ performance bound: a Riccati equation approach. IEEE Trans Autom Control 34(3):293–305

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Doyle J (1978) Guaranteed margins for LQG regulators. IEEE Trans Autom Control 23(4):756–757

Doyle J, Zhou K, Glover K, Bodenheimer B (1994) Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance objectives II: optimal control. IEEE Trans Autom Control 39(8):1575–1587

Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Texts in applied mathematics, vol 36. Springer, New York

Feron E (1997) Analysis of robust $\mathcal{H}_2$ performance using multiplier theory. SIAM J Control Optim 35(1):160–177

Khargonekar P, Rotea M (1991) Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: a convex optimization approach. IEEE Trans Autom Control 36(7):824–837

Oksendal B (1985) Stochastic differential equations. Springer, New York

Paganini F (1999) Convex methods for robust $\mathcal{H}_2$ analysis of continuous time systems. IEEE Trans Autom Control 44(2):239–252

Paganini F, Feron E (1999) LMI methods for robust $\mathcal{H}_2$ analysis: a survey with comparisons. In: El Ghaoui L, Niculescu S (Eds) Recent advances on LMI methods in control. SIAM, Philadelphia

Sánchez-Peña R, Sznaier M (1998) Robust systems theory and applications. Wiley, New York

Scherer C, Gahinet P, Chilali M (1997) Multiobjective output-feedback control via LMI-optimization. IEEE Trans Autom Control 42:896–911

Stoorvogel AA (1993) The robust $\mathcal{H}_2$ control problem: a worst-case design. IEEE Trans Autom Control 38(9):1358–1370

Sznaier M, Rotstein H, Bu J, Sideris A (2000) An exact solution to continuous-time mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control problems. IEEE Trans Autom Control 45(11):2095–2101

Sznaier M, Amishima T, Parrilo PA, Tierno J (2002) A convex approach to robust $\mathcal{H}_2$ performance analysis. Automatica 38:957–966

Zhou K, Glover K, Bodenheimer B, Doyle J (1994) Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ performance objectives I: robust performance analysis. IEEE Trans Autom Control 39(8):1564–1574

Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Robust Model-Predictive Control

Saša Raković
Oxford University, Oxford, UK

Abstract

Model-predictive control (MPC) is indisputably one of the rare modern control techniques that has significantly affected control engineering practice due to its unique ability to systematically handle constraints and optimize performance. Robust MPC (RMPC) is an improved form of the nominal MPC that is intrinsically robust in the face of uncertainty. The main objective of RMPC is to devise an optimization-based control synthesis method that accounts for the intricate interactions of the uncertainty with the system, constraints, and performance criteria in a theoretically rigorous and computationally tractable way. RMPC has become an area of theoretical relevance and practical importance but still offers the fundamental challenge of reaching a meaningful compromise between the quality of structural properties and the computational complexity.

Keywords

Introduction

RMPC is an optimization-based approach to the synthesis of robust control laws for constrained control systems subject to bounded uncertainty. RMPC synthesis can be seen as an adequately defined repetitive decision-making process, in which the underlying decision-making process is a suitably formulated robust optimal control (ROC) problem. The underlying ROC problem is specified in such a way so as to ensure that all possible predictions of the controlled state and corresponding control actions sequences satisfy constraints and that the "worst-case" cost is minimized. The decision variable in the corresponding ROC problem is a control policy (i.e., a sequence of control laws) ensuring that different control actions are allowed at different predicted states, while the uncertainty takes on a role of the adversary. RMPC utilizes recursively the solution to the associated ROC problem in order to implement the feedback control law that is, in fact, equal to the first control law of an optimal control policy.

A theoretically rigorous approach to RMPC synthesis can be obtained either by employing, in a repetitive fashion, the dynamic programming solution of the corresponding ROC problem or by solving online, in a recursive manner, an infinite-dimensional optimization problem (Rawlings and Mayne 2009). In either case, the associated computational complexity renders the exact RMPC synthesis hardly ever tractable. This computational impracticability of the theoretically exact RMPC, in conjunction with the convoluted interactions of the uncertainty with the evolution of the controlled system, constraints, and control objectives, has made RMPC an extremely challenging and active research field. It has become evident that a prominent challenge is to develop a form of RMPC synthesis that adequately handles

the effects of the uncertainty and yet is computationally plausible. Contemporary research proposals aim to address the inevitable trade-off between the quality of guaranteed structural properties and the corresponding computational complexity. A categorization of the existing proposals for RMPC synthesis can be based on the treatment of the effects of the uncertainty. In this sense, two alternative approaches to RMPC synthesis appear to be dominant.

The first category of the alternative approaches is represented by the methods that utilize, when possible, inherent robustness of nominal MPC synthesis. These proposals deploy a nominal MPC, albeit designed for a suitably modified control system, constraints, and control objectives. Such approaches are computationally practicable. However, the effects of the uncertainty are taken care of in an indirect way; the robustness properties of the controlled dynamics are frequently addressed via an *a posteriori* input-to-state stability analysis, which might be unnecessarily conservative and geometrically insensitive. Equally important drawbacks of these approaches to RMPC synthesis arise due to the fact that the nominal MPC synthesis is itself an inherently fragile (nonrobust) process; in particular, the stability property of the conventional MPC might fail to be robust (Grimm et al. 2004) and, furthermore, the optimal control of constrained discrete time systems, employed for the nominal MPC synthesis, can be a fragile process itself (Raković 2009).

The second category of RMPC design methods encapsulates the approaches that take the effects of the uncertainty into account more directly. These proposals are compatible with the emerging consensus: there is a need for the deployment of the simplifying approximations of the underlying control policy and sensible prioritization and modification of control objectives so as to simultaneously enhance computational tractability and ensure *a priori* guarantees of the desirable topological properties and system-theoretic rigor. The simplifying parameterizations of the control policy are employed primarily to allow for a computationally efficient handling of the interactions of the uncertainty with the evolution of the controlled system and constraints. The control objectives are prioritized and modified when necessary, in order to ensure that the corresponding ROC problem is computationally tractable. The effectiveness of such methods depends crucially on the ability to detect a sufficiently rich parameterization of control policy and to devise a systematic way for meaningful simplification of control objectives.

In a stark contrast to a well-matured theory of the nominal MPC synthesis, a systematic assessment of, and unified exposure to, the current state of affairs in the RMPC field is a highly demanding chore. Nevertheless, it is possible to outline the main aspects of the exact RMPC synthesis and to provide an overview of the dominant simplifying approximations.

Contemporary Setting and Uncertainty Effect

The contemporary approach to the exact RMPC synthesis is now delineated in a step-by-step manner.

The system: The most common setting in RMPC synthesis considers the control systems modelled, in discrete time, by

$$x^+ = f(x, u, w), \qquad (1)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $\mathrm{w} \in \mathbb{R}^p$, and $x^+ \in \mathbb{R}^n$ are, respectively, the current state, control and uncertainty, and the successor state, while $f(\cdot, \cdot, \cdot) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$ is the state transition map assumed to be continuous. Thus, when x_k, u_k, and w_k are the state, the control, and the uncertainty at the time instance k, then $x_{k+1} = f(x_k, u_k, w_k)$ is the state at the time instance $k + 1$.

The constraints: The system variables x, u, and w are subject to hard constraints:

$$(x, u, w) \in \mathbb{X} \times \mathbb{U} \times \mathbb{W}, \qquad (2)$$

where the constraint sets $\mathbb{X}$ and $\mathbb{U}$ represent state and control constraints, while the constraint set $\mathbb{W}$ specifies geometric bounds on the uncertainty. The constraint sets $\mathbb{X} \subset \mathbb{R}^n, \mathbb{U} \subset \mathbb{R}^m$, and $\mathbb{W} \subset \mathbb{R}^p$ are assumed to be compact.

The control policy: It is necessary to specify, in a manner that is compatible with the type and nature of the uncertainty, the information available for the RMPC synthesis. The traditional state feedback setting treats the case in which, at any time instance k, the state x_k is known when the current control u_k is determined, while the values of the current and future uncertainty (w_{k+i}) are not known but are guaranteed to take the values within the uncertainty constraint set $\mathbb{W}$ (i.e., $w_{k+i} \in \mathbb{W}$). Within this setting, the use of a control policy,

$$\Pi_{N-1} := \{\pi_0(\cdot), \pi_1(\cdot), \ldots, \pi_{N-1}(\cdot)\}, \quad (3)$$

where N is the prediction horizon and each $\pi_k(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is a control law, is structurally permissible and desirable.

The generalized state and control predictions: Because of the uncertainty, the ordinary state and control predictions, as employed in the nominal MPC, are not suitable. Clearly, when x and $\kappa(x)$ are the current state and control, then the successor state x^+ can take any value in the possible set of successor states $\{f(x, \kappa(x), w) : w \in \mathbb{W}\}$. Consequently, it is necessary to consider suitably generalized state and control predictions. The interaction of the uncertainty with the predicted behavior of the system is captured naturally by invoking the maps $F(\cdot, \cdot)$ and $G(\cdot, \cdot)$ specified, for any subset X of $\mathbb{R}^n$ and any control function $\kappa(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, by

$$F(X, \kappa) := \{f(x, \kappa(x), w) : x \in X, w \in \mathbb{W}\} \text{ and}$$
$$G(X, \kappa) := \{\kappa(x) : x \in X\}. \quad (4)$$

Within the considered setting, the corresponding state and control predictions are, in fact, set-valued and, for each relevant k, obey the relations

$$X_{k+1} = F(X_k, \pi_k) \text{ and } U_k = G(X_k, \pi_k), \text{ with}$$
$$X_0 := \{x\}. \quad (5)$$

The set sequences $\mathbf{X}_N := \{X_0, X_1, \ldots, X_{N-1}, X_N\}$ and $\mathbf{U}_{N-1} := \{U_0, U_1, \ldots, U_{N-1}\}$ represent the possible sets of the predicted states and control actions, which are commonly known as the state and control tubes. Evidently, the state and control tubes are functions of the initial state x and a control policy Π_{N-1}. Reversely, for a given initial state x, any structurally permissible control policy Π_{N-1} results in the possible sets of the predicted states and control actions.

The robust constraint satisfaction: One of the primary objectives in RMPC synthesis is to ensure that the generalized state and control predictions satisfy state and control constraints. Because of the repetitive nature of RMPC, it would be ideal to consider the control policy and generalized state and control predictions over the infinite horizon (i.e., for $N = \infty$). Unfortunately, this is hardly ever practicable in a direct fashion. When the prediction horizon is finite, the robust constraint satisfaction reduces to the conditions that for all $k = 0, 1, \ldots, N-1$, the set inclusions

$$X_k \subseteq \mathbb{X} \text{ and } U_k \subseteq \mathbb{U} \quad (6)$$

hold true and that the possible set of states X_N at the prediction time instance N satisfies the set inclusion

$$X_N \subseteq \mathbb{X}_f, \quad (7)$$

where $\mathbb{X}_f \subseteq \mathbb{X}$ is a suitable terminal constraint set.

The terminal constraint set: In order to account for the utilization of the control policy Π_{N-1} and generalized state and control predictions over the finite horizon N and to ensure that these can be prolonged indirectly over the infinite horizon, a terminal constraint set is employed. This set is obtained by considering the uncertain dynamics

$$x^+ = f(x, \kappa_f(x), w) \quad (8)$$

controlled by a local control function $\kappa_f(\cdot)$. The design of a control law $\kappa_f(\cdot)$ is usually performed offline in an optimal manner by considering the unconstrained version of the system (1), while the terminal constraint set $\mathbb{X}_f$ accounts locally for the state and control constraints. The terminal constraint set $\mathbb{X}_f$ is assumed to be compact and robust positively invariant for the dynamics (8) and constraint sets (2). Thus, the set $\mathbb{X}_f$ and a local control function $\kappa_f(\cdot)$ satisfy

$$F\left(\mathbb{X}_f, \kappa_f\right) \subseteq \mathbb{X}_f \subseteq \mathbb{X} \text{ and } \mathbb{U}_f$$
$$:= G\left(\mathbb{X}_f, \kappa_f\right) \subseteq \mathbb{U}, \qquad (9)$$

or, equivalently, $\mathbb{X}_f \subseteq \mathbb{X}$, and for all $x \in \mathbb{X}_f$, it holds that $\kappa_f(x) \in \mathbb{U}$ and $\forall w \in \mathbb{W}$, $f(x, \kappa_f(x), w) \in \mathbb{X}_f$. The most appropriate choice for $\mathbb{X}_f$ is the maximal robust positively invariant set for the dynamics (8) and constraint sets (2).

The generalized origin: Due to the presence of the uncertainty, the stabilization of the origin might not be attainable and, thus, it might be necessary to consider the origin in a generalized sense. The most natural candidate for the generalized origin is a minimal robust positively invariant set for the dynamics (8) and constraint sets (2). This set is entirely determined by the associated state set dynamics

$$X^+ = F\left(X, \kappa_f\right), \qquad (10)$$

which are completely induced by the local dynamics (8) and the uncertainty constraint set $\mathbb{W}$. The generalized origin, namely, the minimal robust positively invariant set, is compact and well defined in the case when the local control function $\kappa_f(\cdot)$ ensures that the corresponding map $F(\cdot, \kappa_f)$ is a contraction on the space of compact subsets of $\mathbb{X}_f$ (Artstein and Raković 2008), which we assume to be the case. The generalized origin $\mathbb{X}_{\mathcal{O}}$ is the unique solution to the fixed-point set equation.

$$X = F\left(X, \kappa_f\right), \qquad (11)$$

and is an exponentially stable attractor for the state set dynamics (10) with the basin of attraction being the space of compact subsets of $\mathbb{X}_f$. Thus, the conventional (0,0) fixed-point pair ought to be replaced by the fixed-point pair of sets $(\mathbb{X}_{\mathcal{O}}, \mathbb{U}_{\mathcal{O}})$ required to satisfy

$$\mathbb{X}_{\mathcal{O}} = F\left(\mathbb{X}_{\mathcal{O}}, \kappa_f\right) \subseteq \text{interior}\left(\mathbb{X}_f\right) \text{ and}$$
$$\mathbb{U}_{\mathcal{O}} := G\left(\mathbb{X}_{\mathcal{O}}, \kappa_f\right) \subseteq \mathbb{U}_f. \qquad (12)$$

The generalized cost functions: The performance requirements are, as usual, expressed via a cost function, which is obtained by considering a stage cost function $\ell(\cdot, \cdot) : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_+$ and a terminal cost function $V_f(\cdot) : \mathbb{X}_f \to \mathbb{R}_+$. The stage cost function $\ell(\cdot, \cdot)$ is continuous and, due to the uncertainty, adequately lower bounded w.r.t. to the generalized origin $\mathbb{X}_{\mathcal{O}}$. The latter condition requires that for all $x \in \mathbb{X}$ and all $u \in \mathbb{U}$, the function $\ell(\cdot, \cdot)$ satisfies

$$\alpha_1\left(\text{dist}\left(\mathbb{X}_{\mathcal{O}}, x\right)\right) \le \ell\left(x, u\right), \qquad (13)$$

where $\alpha_1(\cdot)$ is a $\mathcal{K}$-class (Kamke's) function and $\text{dist}(\mathbb{X}_{\mathcal{O}}, \cdot)$ is the distance function from the set $\mathbb{X}_{\mathcal{O}}$. The consideration of the generalized origin requires the additional condition that for all $x \in \mathbb{X}_{\mathcal{O}}$, the use of local control function $\kappa_f(\cdot)$ is "free of charge" w.r.t. $\ell(\cdot, \cdot)$, i.e., that for all $x \in \mathbb{X}_{\mathcal{O}}$, we have

$$\ell\left(x, \kappa_f\left(x\right)\right) = 0. \qquad (14)$$

As in the case of the terminal constraint set $\mathbb{X}_f$, the terminal cost function $V_f(\cdot)$ is employed to account for the utilization of the finite prediction horizon N, and it should provide locally a theoretically suitable upper bound of the highly desired infinite horizon cost. The terminal cost function $V_f(\cdot)$ is assumed to be continuous and adequately upper bounded w.r.t. the generalized origin $\mathbb{X}_{\mathcal{O}}$. The latter bound reduces to the requirement that for all $x \in \mathbb{X}_f$, we have

$$V_f(x) \le \alpha_2\left(\text{dist}\left(\mathbb{X}_{\mathcal{O}}, x\right)\right), \qquad (15)$$

where, as above, $\alpha_2(\cdot)$ is a $\mathcal{K}$-class function. In addition, the terminal cost function $V_f(\cdot)$ satisfies locally a usual condition for robust stabilization,

which is expressed by the requirement that for all $x \in \mathbb{X}_f$ and all $w \in \mathbb{W}$, it holds that

$$V_f\left(f\left(x, \kappa_f\left(x\right), w\right)\right) - V_f\left(x\right) \leq -\ell\left(x, \kappa_f\left(x\right)\right). \tag{16}$$

The cost function $V_N(\cdot, \cdot, \cdot)$ is defined, for all $x \in \mathbb{X}$, all Π_{N-1}, and all $\mathbf{w}_{N-1} := \{w_0, w_1, \ldots, w_{N-1}\}$, by

$$V_N\left(x, \Pi_{N-1}, \mathbf{w}_{N-1}\right) := \sum_{k=0}^{N-1} \ell\left(x_k, u_k\right) + V_f\left(x_N\right), \tag{17}$$

where, for notational simplicity, $u_k := \pi_k(x_k)$ and $x_k := x_k(x, \Pi_{N-1}, \mathbf{w}_{N-1})$ denote the solution of (1) when the initial state is x, control policy is Π_{N-1}, and uncertainty realization is $\mathbf{w}_{N-1}$.

The exact ROC: In view of the uncertainty, the corresponding exact ROC problem $\mathbb{P}_N(x)$, for any $x \in \mathbb{X}$, aims to optimize the "worst-case" performance so that it takes the form of an infinite-dimensional minimaximization:

$$J_N\left(x, \Pi_{N-1}\right) := \max_{\mathbf{w}_{N-1} \in \mathbb{W}^N} V_N\left(x, \Pi_{N-1}, \mathbf{w}_{N-1}\right),$$

$$V_N^0\left(x\right) := \min_{\Pi_{N-1} \in \Pi_{N-1}(x)} J_N\left(x, \Pi_{N-1}\right),$$

$$\Pi_{N-1}^0\left(x\right) \in \arg \min_{\Pi_{N-1} \in \Pi_{N-1}(x)} J_N\left(x, \Pi_{N-1}\right), \tag{18}$$

where $\mathbf{\Pi}_{N-1}(x)$ denotes the set of the constraint admissible control policies defined, for all $x \in \mathbb{X}$, by

$$\Pi_{N-1}\left(x\right) := \{\Pi_{N-1} : \text{conditions (5)–(7) hold}\}. \tag{19}$$

The value function $V_N^0(\cdot)$ might not admit a unique optimal control policy, so that $\Pi_{N-1}^0(\cdot)$ represents a selection from the set of optimal control policies (this selection is usually induced by a numerical solver employed for the online calculations). The effective domain $\mathcal{X}_N$ of the value function $V_N^0(\cdot)$ and associated optimal control policy $\Pi_{N-1}^0(\cdot)$ is given by

$$\mathcal{X}_N := \{x \in \mathbb{R}^n : \Pi_{N-1}(x) \neq \emptyset\}. \tag{20}$$

and is known in the literature as the N-step min–max controllable set to a target set $\mathbb{X}_f$. Within the considered setting, the set $\mathcal{X}_N$ is a compact subset of $\mathbb{X}$ such that $\mathbb{X}_f \subseteq \mathcal{X}_N$.

The exact RMPC: The exact RMPC synthesis requires online solution of the minimaximization (18) in order to implement numerically the control law $\pi_0^0(\cdot)$. The control law $\pi_0^0(\cdot)$ is well defined for all $x \in \mathcal{X}_N$, and it induces the controlled uncertain dynamics specified, for all $x \in \mathcal{X}_N$, by

$$x^+ \in \mathcal{F}\left(x\right), \mathcal{F}\left(x\right) := \{f\left(x, \pi_0^0\left(x\right), w\right) : w \in \mathbb{W}\}. \tag{21}$$

Within the considered setting, the exact RMPC law $\pi_0^0(\cdot)$ renders the N-step min–max controllable set $\mathcal{X}_N$ robust positively invariant. Namely, for all $x \in \mathcal{X}_N$, it holds that

$$\mathcal{F}\left(x\right) \subseteq \mathcal{X}_N \subseteq \mathbb{X} \text{ and } \pi_0^0\left(x\right) \in \mathbb{U}. \tag{22}$$

Furthermore, the associated value function $V_N^0(\cdot) : \mathcal{X}_N \rightarrow \mathbb{R}_+$ is, by construction, a Lyapunov certificate verifying the robust asymptotic stability of the generalized origin $\mathbb{X}_{\mathcal{O}}$ for the controlled uncertain dynamics (21) with the basin of attraction being equal to the N-step min–max controllable set $\mathcal{X}_N$. More precisely, for all $x \in \mathcal{X}_N$, it holds that

$$\alpha_1\left(\text{dist}\left(\mathbb{X}_{\mathcal{O}}, x\right)\right) \leq V_N^0\left(x\right) \leq \alpha_3\left(\text{dist}\left(\mathbb{X}_{\mathcal{O}}, x\right)\right), \tag{23}$$

where $\alpha_3(\cdot)$ is a suitable $\mathcal{K}$-class function, while for all $x \in \mathcal{X}_N$ and all $x^+ \in \mathcal{F}(x)$, it holds that

$$V_N^0\left(x^+\right) - V_N^0\left(x\right) \leq -\alpha_1\left(\text{dist}\left(\mathbb{X}_{\mathcal{O}}, x\right)\right). \tag{24}$$

Clearly, under fairly natural conditions, the exact RMPC synthesis induces rather strong structural properties, but the associated computational complexity is overwhelming. However, in the above overview, the effects of the uncertainty have been "dissected" and the "basic building blocks" employed for the exact RMPC synthesis have been clearly identified. In turn, this step-by-step overview suggests indirectly the meaningful

and simplifying approximations in order to enhance computational practicability.

Computational Simplifications

The computational intractability of the exact RMPC synthesis can be tackled by considering suitable parameterizations of control policy Π_{N-1} and associated state and control tubes $\mathbf{X}_N$ and $\mathbf{U}_{N-1}$ and by adopting computationally simpler performance criteria.

The core simplification is the use of finite-dimensional parameterization of control policy. The control policy should be suitably parameterized so as to allow for the utilization of both the least conservative generalized state and control predictions and a range of simpler, but sensible, cost functions.

The explicit form of the exact state and control tubes is usually highly complex, and it is computationally beneficial to employ, when feasible, the implicit representation of the possible sets of predicted state and control actions. An alternative is to utilize outer-bounding approximations of the exact state and control tubes; these are obtained by making use of simpler sets that usually admit finite-dimensional parameterizations. In the latter case, the exact set dynamics of the state and control tubes given by (5) are usually relaxed to set inclusions

$$\{x_0\} \subseteq X_0, \text{ and, } F(X_k, \pi_k) \subseteq X_{k+1}$$
$$\text{and } G(X_k, \pi_k) \subseteq U_k.$$

The generalized origin, i.e., the minimal robust positively invariant set $\mathbb{X}_{\mathcal{O}}$, is an integral component for the analysis. Its explicit computation is rather demanding and, hence, its use for the online calculations might not be convenient. A computationally feasible alternative is to deploy the terminal constraint set $\mathbb{X}_f$ as a "relaxed form" of the generalized origin; this is particularly beneficial when the local control function $\kappa_f(\cdot)$ is optimal w.r.t. infinite horizon cost associated with the unconstrained version of the system (1).

The performance requirements should be carefully prioritized and modified when necessary, in such a way so as to be expressible by the cost functions that do not require intractable minimax optimization but still ensure that the associated value function verifies the robust stability and attractivity of the generalized origin $\mathbb{X}_{\mathcal{O}}$ or the terminal constraint set $\mathbb{X}_f$.

The outlined guidelines have played a pivotal role in devising a number of theoretically sound and computationally efficient parameterized RMPC syntheses within the setting of linear control systems subject to additive disturbances and polytopic constraints. In this linear–polytopic setting, the state transition map $f(\cdot, \cdot, \cdot)$ of (1) is linear:

$$f(x, u, w) = Ax + Bu + w, \qquad (25)$$

where the matrix pair $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$ is assumed to be known and strictly stabilizable. The local control function $\kappa_f(\cdot)$ and associated local uncertain dynamics are linear:

$$u = Kx \text{ and } x^+ = (A + BK)x + w. \quad (26)$$

The matrix $K \in \mathbb{R}^{m \times n}$ is designed offline and is such that the eigenvalues of the matrix $A + BK$ are strictly inside of the unit circle. The constraint sets $\mathbb{X}$ and $\mathbb{U}$ are polytopes (A polytope is a convex and compact set specified by finitely many linear/affine inequalities, or by a convex hull of finitely many points) in $\mathbb{R}^n$ and $\mathbb{R}^m$ that contain the origin in their interior. The uncertainty constraint set $\mathbb{W}$ is a polytope in $\mathbb{R}^n$ that contains the origin.

The terminal constraint set $\mathbb{X}_f$ is the maximal robust positively invariant for $x^+ = (A+BK)x + w$ and constraint set $(\mathbb{X}_K, \mathbb{W})$ where $\mathbb{X}_K := \{x \in \mathbb{X} : Kx \in \mathbb{U}\}$. The set $\mathbb{X}_f$ is assumed to be a polytope in $\mathbb{R}^n$ that contains the generalized origin $\mathbb{X}_{\mathcal{O}}$ (which is the minimal robust positively invariant set for $x^+ = (A + BK)x + w$ and constraint set $(\mathbb{X}_K, \mathbb{W})$) in its interior.

It has recently been demonstrated that the major simplified RMPC syntheses in the linear–polytopic setting employ control policies within the class of separable state feedback (SSF) control policies (Raković 2012). More precisely, the predictions of the overall states x_k and

associated control actions u_k are parameterized in terms of the predictions of the partial states $x_{(j,k)}$, $j = 0, 1, \ldots, k$ and partial control actions $u_{(j,k)}$, $j = 0, 1, \ldots, k$ via

$$x_k = \sum_{j=0}^{k} x_{(j,k)} \text{ and } u_k = \sum_{j=0}^{k} u_{(j,k)}, \qquad (27)$$

where, for notational simplicity, $u_k := \pi_k(x_k)$ and $u_{(j,k)} := \pi_{(j,k)}(x_{(j,k)})$. To ensure the dynamical consistency with (25), the predicted partial states $x_{(j,k)}$ evolve according to

$$x_{(j,k+1)} = Ax_{(j,k)} + Bu_{(j,k)}, \qquad (28)$$

(for $j = 0, 1, \ldots, N - 1$ and $k = j, j + 1 \ldots, N - 1$), while the "partial" initial conditions $x_{(k,k)}$ satisfy

$$x_{(0,0)} = x \text{ and} \qquad (29a)$$

$$x_{(k,k)} = w_{k-1} \text{ for } k = 1, 2, \ldots, N. \qquad (29b)$$

As elaborated on in Raković (2012) and Raković et al. (2012), the utilization of the SSF control policy allows for:

- The deployment of the highly desirable implicit representation of the exact state and control tubes induced by the SSF control policy. This implicit representation is parameterized via $O(N^2)$ decision variables.
- The numerically convenient formulation of the robust constraint satisfaction via $O(N^2)$ linear/affine inequalities and equalities.
- The computationally efficient minimization of an upper bound of the "worst-case" cost for which the stage and terminal cost functions are specified in terms of the weighted distances from the terminal constraint set $\mathbb{X}_f$ and the associated control set $\mathbb{U}_f = K\mathbb{X}_f$.

As shown in Raković (2012) and Raković et al. (2012), the RMPC control laws, based on the use of the SSF control policy, can be implemented online by solving a standard convex optimization problem whose complexity (in terms of the numbers of decision variables and affine inequalities and equalities) is $O(N^2)$. The corresponding

RMPC synthesis ensures directly that the terminal constraint set $\mathbb{X}_f$ is robustly exponentially stable, and it also induces indirectly the robust exponential stability of the generalized origin $\mathbb{X}_{\mathcal{O}}$.

The previously dominant control policy parameterizations include time-invariant affine state feedback (TIASF), time-varying affine state feedback (TVASF), and affine in the past disturbances feedback (APDF) control policies. All of these parameterizations are subsumed by the SSF control policy, as all of them induce additional structural restrictions on the parameterizations of the predicted state and control actions specified in (27) and on the associated dynamics given by (28), (29) and (30). In particular, the TIASF control policy (Chisci et al. 2001; Gossner et al. 1997) imposes structural restrictions that, for each relevant k,

$$u_{(j,k)} = Kx_{(j,k)} \text{ for } j = 1, 2, \ldots, k, \qquad (30)$$

where K is the local control matrix of (26). The TVASF control policy (Löfberg 2003) induces less restrictive requirements that, for each relevant k,

$$u_{(j,k)} = K_{(j,k)}x_{(j,j)} \text{ for } j = 1, 2, \ldots, k, \qquad (31)$$

where the matrices $K_{(j,k)} \in \mathbb{R}^{m \times n}$ are part of the decision variable. The APDF control policy (Goulart et al. 2006; Löfberg 2003) is an algebraic reparameterization of the TVASF control policy, which requires the conditions that, for each relevant k,

$$u_{(j,k)} = M_{(j,k)}x_{(j,k)} \text{ for } j = 1, 2, \ldots, k, \qquad (32)$$

where the matrices $M_{(j,k)} \in \mathbb{R}^{m \times n}$ are part of the decision variable. A comprehensive trade-off analysis between the quality of guaranteed structural properties and the associated computational complexity and a theoretically meaningful ranking of the existing RMPC syntheses in the linear–polytopic setting is reported in the recent plenary paper (Raković 2012). Therein, it is demonstrated that the dominant approach is the RMPC synthesis utilizing the SSF control policy

(Raković 2012) (also known as the parameterized tube MPC (Raković et al. 2012)).

Summary and Future Directions

The exact RMPC synthesis has reached a remarkable degree of theoretical maturity in the general setting. The corresponding theoretical advances are, however, accompanied with the impeding computational complexity. On the bright side of the things, a number of rather sophisticated RMPC synthesis methods, which are both computationally efficient and theoretically sound, have been developed for the frequently encountered linear–polytopic case.

The further advances in the RMPC field might be driven by the utilization of more structured types and models of the uncertainty. The challenge of devising a computationally efficient and theoretically sound RMPC synthesis might need to be tackled in several phases; the initial steps might focus on adequate RMPC synthesis for particular classes of nonlinear control systems. Finally, it would seem reasonable to expect that the lessons learned in the RMPC field might play an important role for the research developments in the fields of the stochastic and adaptive MPC.

Cross-References

▶ Nominal Model-Predictive Control
▶ Stochastic Model Predictive Control

Recommended Reading

The recent monograph (Rawlings and Mayne (2009)) provides an in-depth systematic exposure to the RMPC field and is also a rich source of relevant references. The invaluable overview of the theory and computations of the maximal and minimal robust positively invariant sets can be found in (Artstein and Raković (2008), Kolmanovsky and Gilbert (1998), Raković et al. (2005), and Blanchini and Miani (2008)). The important paper (Scokaert and Mayne (1998))

points out the theoretical benefits of the use of the control policy, but it also indicates indirectly the computational impracticability of the associated feedback min–max RMPC. The early tube MPC synthesis (Mayne et al. 2005) is both computationally efficient and theoretically sound, and it represents an important step forward in the linear–polytopic setting. The so-called homothetic tube MPC synthesis (Raković et al. 2013) is a recent improvement of the first generation of the tube MPC synthesis (Mayne et al. 2005), and it has a high potential to effectively handle the parametric uncertainty of the matrix pair (A, B). The current state of the art in the linear–polytopic setting is reached by the RMPC synthesis using the SSF control policy (Raković 2012; Raković et al. 2012). The output feedback RMPC synthesis in the linear–polytopic setting can be handled with direct extensions of the tube MPC syntheses (Mayne et al. 2009).

Bibliography

Artstein Z, Raković SV (2008) Feedback and invariance under uncertainty via set iterates. Automatica 44(2):520–525

Blanchini F, Miani S (2008) Set–theoretic methods in control. Systems & control: foundations & applications. Birkhauser, Boston/Basel/Berlin

Chisci L, Rossiter JA, Zappa G (2001) Systems with persistent disturbances: predictive control with restricted constraints. Automatica 37:1019–1028

Gossner JR, Kouvaritakis B, Rossiter JA (1997) Stable generalised predictive control in the presence of constraints and bounded disturbances. Automatica 33(4):551–568

Goulart PJ, Kerrigan EC, Maciejowski JM (2006) Optimization over state feedback policies for robust control with constraints. Automatica 42(4):523–533

Grimm G, Messina MJ, Tuna SE, Teel AR (2004) Examples when nonlinear model predictive control is nonrobust. Automatica 40:1729–1738

Kolmanovsky IV, Gilbert EG (1998) Theory and computation of disturbance invariant sets for discrete time linear systems. Math Problems Eng Theory Methods Appl 4:317–367

Löfberg J (2003) Minimax approaches to robust model predictive control. Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Linköping

Mayne DQ, Seron M, Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. Automatica 41:219–224

Mayne DQ, Raković SV, Findeisen R, Allgöwer F (2009) Robust output feedback model predictive control of constrained linear systems: time varying case. Automatica 45:2082–2087

Raković SV (2009) Set theoretic methods in model predictive control. In: Lalo Magni, Davide Martino Raimondo and Frank Allgöwer (eds) Nonlinear model predictive control: towards new challenging applications. Lecture notes in control and information sciences, vol 384. Springer, Berlin/Heidelberg, pp 41–54

Raković SV (2012) Invention of prediction structures and categorization of robust MPC syntheses. In: Proceedings of the 4th IFAC conference on nonlinear model predictive control NMPC 2012, Noordwijkerhout. Plenary Paper, 245–273

Raković SV, Kerrigan EC, Kouramas KI, Mayne DQ (2005) Invariant approximations of the minimal robustly positively invariant set. IEEE Trans Autom Control 50(3):406–410

Raković SV, Kouvaritakis B, Cannon M, Panos C, Findeisen R (2012) Parameterized tube model predictive control. IEEE Trans Autom Control 57(11):2746–2761

Raković SV, Kouvaritakis B, Cannon M (2013) Equinormalization and exact scaling dynamics in homothetic tube model predictive control. Syst Control Lett 62(2):209–217

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design. Nob Hill Publishing, Madison

Scokaert POM, Mayne DQ (1998) Min–max feedback model predictive control for constrained linear systems. IEEE Trans Autom Control 43:1136–1142

Robust Synthesis and Robustness Analysis Techniques and Tools

Gary Balas[1], Andrew Packard[2], and Peter Seiler[1]
[1]Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA
[2]Mechanical Engineering Department, University of California, Berkeley, CA, USA

Abstract

This entry provides a brief summary of the synthesis and analysis tools that have been developed by the robust control community. Many software tools have been developed to implement the ma-

jor theoretical techniques in robust control. These software tools have enabled robust synthesis and analysis techniques to be successfully applied to numerous industrial applications.

Keywords

Integral quadratic constraint; Linear matrix inequality; Model uncertainty; Robust control toolbox; Structured singular values

Introduction

Robust control is a methodology to address the effect of uncertainty on feedback systems. This approach includes techniques and tools to model system uncertainty, assess stability and/or performance characteristics of the uncertain system, and synthesize controllers for uncertain systems. The theory was developed over a number of years. The foundational results can be found in classical papers Packard and Doyle (1993a), Desoer et al. (1980), Doyle (1978, 1982), Doyle et al. (1989), Doyle and Stein (1981), Megretski and Rantzer (1997), Safonov (1982), Willems (1971), and Zames (1981) and more recent textbooks Boyd et al. (1994), Desoer and Vidyasagar (2008), Dullerud and Paganini (2000), Francis (1987), Skogestad and Postlethwaite (2005), Vidyasagar (1985), and Zhou et al. (1996). It should be emphasized that this entry is not meant to be a survey and more complete references to the literature can be found in the cited textbooks. The remainder of this entry discusses the main theoretical and computational tools for robust synthesis and robustness analysis.

Notation

$\mathbf{R}$ and $\mathbf{C}$ denote the set of real and complex numbers, respectively. $\mathbf{R}^{m \times n}$ and $\mathbf{C}^{m \times n}$ denote the sets of $m \times n$ matrices whose elements are in $\mathbf{R}$ and $\mathbf{C}$, respectively. A single superscript index is used for vectors, e.g., $\mathbf{R}^n$ denotes the set of $n \times 1$ vectors whose elements are in $\mathbf{R}$. For a

matrix $M \in \mathbf{C}^{m \times n}$, M^T denotes the transpose and M^* denotes the complex conjugate transpose. A matrix M is Hermitian (Skew-Hermitian) if $M = M^*$ ($M = -M^*$). The maximum singular value of a matrix M is denoted by $\bar{\sigma}(M)$. The trace of a matrix M, denoted $tr[M]$, is the sum of the diagonal elements. $M = M^*$ is a positive semidefinite matrix, denoted $M \succeq 0$, if all eigenvalues are nonnegative. $M = M^*$ is negative semidefinite, denoted $M \preceq 0$, if $-M \succeq 0$. $\mathcal{L}_2^n[0, \infty)$ is the space of functions $u : [0, \infty) \to \mathbf{R}^n$ satisfying $\|u\| < \infty$ where $\|u\| := \left[\int_0^\infty u(t)^T u(t) \, dt \right]^{0.5}$. For $u \in \mathcal{L}_2^n[0, \infty)$, u_T denotes the truncated function $u_T(t) = u(t)$ for $t \leq T$ and $u(t) = 0$ otherwise. The extended space, denoted $\mathcal{L}_{2e}$, is the set of functions u such that $u_T \in \mathcal{L}_2$ for all $T \geq 0$. The Fourier transform $\hat{v} := \mathcal{F}(v)$ maps the time domain signal $v \in \mathcal{L}_2^n[0, \infty)$ to the frequency domain by

$$\hat{v}(j\omega) := \int_0^\infty e^{-j\omega t} v(t) dt \qquad (1)$$

Capital letters are used to represent dynamical systems. For linear systems, the same letter is used to represent the system, its convolution kernel, as well as its frequency-response function. Lowercase letters denote time-signals, and ω represents the continuous-time frequency variable. For an $m \times n$ system G, define the H_∞ and H_2 norms as $\|G\|_\infty = \sup_\omega \bar{\sigma}(G(j\omega))$ and $\|G\|_2 = \sqrt{\frac{1}{2\pi} \int_{-\infty}^\infty tr[G(j\omega)^* G(j\omega)] d\omega}$. The $\mathcal{L}_1$ norm of G is defined as $\|G\|_1 = \max_{1 \leq i \leq m} \sum_{j=1}^n \int_0^\infty |g_{ij}(t)| dt$ where $g_{ij}(t)$ is the response of the ith output due to a unit impulse in the jth input. The entry describes continuous-time systems. Most results carry over, in a similar form, to discrete-time systems.

Theoretical Tools

Uncertainty Modeling

In order to analyze and/or design for the degrading effects of uncertainty, it is imperative that explicit models of uncertainty be characterized. Two distinct forms of uncertainty are considered: signal uncertainty and model uncertainty. Signal uncertainty represents external signals (plant disturbances, sensor noise, reference signals) as sets of time functions, with explicit descriptions. For example, a particular reference input might be characterized as belonging to the set $\left\{ \frac{4}{2s+1} d : d \in \mathcal{L}_2, \|d\|_2 \leq 1 \right\}$. This set is often referred to as a *weighted ball in* $\mathcal{L}_2$. The transfer function $\frac{4}{2s+1}$ is called a *weighting function* and it shapes the normalized signals d, in a manner that its output represents the actual traits of the reference inputs that occur in practice.

Model uncertainty represents unknown or partially specified gains (more generally, operators) that relate pairs of signals in the model. For example, z and w are signals within a model and are related by an operator $\mathcal{N}$ as $w = \mathcal{N}(z)$. Typical partial specifications either constrain $\mathcal{N}$ to be drawn from a specified set or describe the set of signals (w, z) that $\mathcal{N}$ allows. An *uncertain parameter* δ is modeled as time-invariant (i.e., constant), belonging to the interval $[a, b]$ and relating z and w as $w(t) = \delta z(t)$. An *uncertain linear dynamic element*, Δ is modeled as linear, time-invariant, causal system, described by a convolution kernel δ whose frequency-response function (i.e., Fourier transform) satisfies $\max_\omega \left| \hat{\delta}(j\omega) \right| \leq 1$, and relating z and w as $w = \delta \star z$. More generally, consider an $\mathcal{L}_2$ bounded, causal operator, mapping $\mathcal{L}_{2,e} \to \mathcal{L}_{2,e}$ relating the signals as $w = \Delta(z)$. The behavior of Δ is unknown but constrained by a family of multipliers, $\{\Pi_\alpha\}_{\alpha \in \mathcal{A}}$. Specifically, each Π_α is a Hermitian, matrix-valued function of frequency, and for any $z \in \mathcal{L}_2$, the mapping Δ is known to satisfy

$$\int_{-\infty}^\infty \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \Pi_\alpha(\omega) \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega \geq 0$$

This is called an *integral quadratic constraint* (IQC) description of Δ, as the input/output pairs of Δ satisfy a family of quadratic, integral constraints. These different descriptions of model uncertainty are related. For example, if $w(t) = \delta z(t)$, with $w(t) \in \mathbf{R}^n$ and $z(t) \in \mathbf{R}^n$, and $\delta \in \mathbf{R}, |\delta| \leq 1$, then for any Hermitian-valued $X : \mathbf{R} \to \mathbf{C}^{n \times n}$ with $X(\omega) \succeq 0$ for all $\omega \in \mathbf{R}$ and Skew-Hermitian $Y : \mathbf{R} \to \mathbf{C}^{n \times n}$,

$$\int_{-\infty}^{\infty} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix}^* \begin{bmatrix} X(\omega) & Y(\omega) \\ Y^*(\omega) & -X(\omega) \end{bmatrix} \begin{bmatrix} \hat{z}(j\omega) \\ \hat{w}(j\omega) \end{bmatrix} d\omega$$
$$= \int_{-\infty}^{\infty} \hat{z}^*(j\omega) \left[(1 - \delta^2) X(\omega) \right] \hat{z}(j\omega) d\omega$$

which is always $\succeq 0$. Hence, the *uncertain parameter* can be recast as an operator satisfying an infinite family of IQCs. Nonlinear operators may also satisfy IQCs and it is common to "model" known nonlinear elements (e.g., saturation) by enumerating IQCs that they satisfy (Megretski and Rantzer 1997). An *uncertain dynamic model* is made up of an interconnection of these uncertain elements with a known (usually) linear system G.

Performance Metric

The main goal of robustness analysis is to assess the degrading effects of uncertainty. For this, a concrete notion of performance is needed, resulting in a mathematical/computational exercise to quantify the average or worst-case effects of the two types of uncertainty, signal and model, described earlier. In the robust control framework, adequate performance is characterized in terms of the variability of possible behavior of particular signals. For instance, in the presence of reference inputs and disturbance inputs, as well as parameter uncertainty, it is required that tracking errors (e) and control inputs (u) remain small. A common measure of smallness is the $\mathcal{L}_2$ norm of signals. Typically, frequency-dependent weighting functions are used to preferentially weight one frequency range over another and/or to weight one signal relative to another. In this way, adequate performance be defined as $\|W \begin{bmatrix} e \\ u \end{bmatrix}\|_2 \leq 1$, where W is a stable, linear system, called the "output" weighting function. Weighting functions are often used to transform a collection of performance objectives into a single norm bound objective in the robust control framework.

Robustness Analysis

Robustness analysis refers to the task of ascertaining the stability and/or performance characteristics of the uncertain system, given the

limited knowledge about the uncertain information. The main result from Megretski and Rantzer (1997) concerns the stability of the interconnection shown in Fig. 1, where G is a known, stable, linear system and Δ is an operator that satisfies the IQC defined by Π. Under some important technical conditions, the theorem states "if there exists an $\epsilon > 0$ such that

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Pi(\omega) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} \preceq -\epsilon I \quad (2)$$

for all $\omega \in \mathbf{R}$, then the interconnection is stable." Stability here refers to finite $\mathcal{L}_2$ gain from inputs (r_1, r_2) to loop signals (u_1, u_2).

Multiple IQCs satisfied by Δ can be incorporated into the analysis. In particular, assume that Δ satisfies the IQCs defined by the multipliers $\{\Pi_k\}_{k=1}^N$. Then Δ satisfies the IQC defined by any multiplier of the form $\Pi^\alpha := \sum_{k=1}^N \alpha_k \Pi_k$ where $\alpha_k \geq 0$. The stability test amounts to a semi-infinite, semidefinite feasibility problem: find nonnegative scalars $\{\alpha_k\}_{k=1}^N$ such that for some $\epsilon > 0$,

$$\begin{bmatrix} G(j\omega) \\ I \end{bmatrix}^* \Pi^\alpha(\omega) \begin{bmatrix} G(j\omega) \\ I \end{bmatrix} \preceq -\epsilon I \quad (3)$$

for all $\omega \in \mathbf{R}$. This infinite family of matrix inequalities (one for each frequency) can be equivalently expressed as a finite-dimensional linear matrix inequality (LMI) under some additional restrictions.

The structured singular value (μ) approach provides an alternative robust stability test in the case of only linear, time-invariant uncertainty
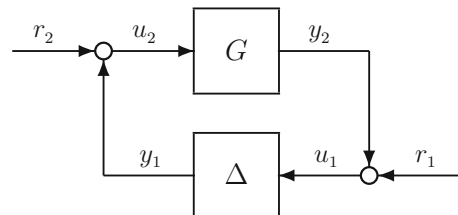
Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 1 Feedback interconnection for IQC stability test

(parametric or dynamic). Suppose Δ is drawn from a set of matrices, $\Delta \subseteq \mathbf{C}^{m \times n}$ of the form

$$\Delta = \big\{ \text{diag} \big[\delta_1^r I_{t_1}, \ldots, \delta_V^r I_{t_V}, \\ \delta_1^c I_{r_1}, \ldots, \delta_S^c I_{r_S}, \Delta_1, \ldots, \Delta_F \big] : \\ \delta_k^r \in \mathbf{R}, \delta_i^c \in \mathbf{C}, \Delta_j \in \mathbf{C}^{m_j \times n_j} \big\}$$

The inclusion of complex-valued, uncertain matrices within Δ may seem unusual and hard to motivate. However, in terms of their effect on stability, these are equivalent to the *uncertain linear dynamic element* introduced earlier in the *Uncertainty Modeling* section. This is discussed in more detail in the entry ▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems.

Using the Nyquist stability criterion, the (G, Δ) interconnection is stable for all $\Delta \in \Delta$, with $\bar{\sigma}(\Delta) < \beta$ if and only if G is stable, and

$$\det(I - G(j\omega)\Delta) \neq 0$$

for all $\Delta \in \Delta$ with $\bar{\sigma}(\Delta) < \beta$ and all $\omega \in \mathbf{R}$ including $\omega = \infty$. The importance of the nonvanishing determinant condition warrants a definition of its own, the *structured singular value*. For a matrix $M \in \mathbf{C}^{n \times m}$, and Δ as given, define

$$\mu_{\Delta}(M) := \frac{1}{\min \{\bar{\sigma}(\Delta) : \Delta \in \Delta, \det(I - M\Delta) = 0\}}$$

unless no $\Delta \in \Delta$ makes $(I - M\Delta)$ singular, then $\mu_{\Delta}(M) := 0$. In this parlance, the (G, Δ) interconnection is stable for all $\Delta \in \Delta$, with $\bar{\sigma}(\Delta) < \beta$ if and only if

$$\mu_{\Delta}(G(j\omega)) \leq \frac{1}{\beta}$$

for all $\omega \in \mathbf{R}$ including $\omega = \infty$.

In summary, the structured singular value approach employs a Nyquist-based argument, resulting in a nonvanishing determinant condition, which must hold over all frequency and all possible frequency-response values of the uncertain elements. However, checking the nonvanishing determinant is difficult, and sufficient conditions,

in the form of semidefinite programs (Doyle 1982; Fan et al. 1991) to ensure this are derived. This results in semidefinite feasibility problems which must hold at all frequencies. It is common to verify these only on a finite grid of frequencies, which is equivalent to ensuring that the closed-loop poles cannot migrate across the stability boundary at these frequencies. Semidefinite programs can be defined which carve out intervals around these fixed frequencies to completely guarantee stability.

Robust Synthesis

Synthesis refers to the mathematical design of the control law. The nominal synthesis problem (with no uncertainty) is formulated using the generic feedback structure shown in Fig. 2. The various signals in the diagram are the control inputs u, measurements y, exogenous disturbances d, and regulated variables e. P is a generalized plant that contains all information required to specify the synthesis problem. This includes the dynamics of the actual plant being controlled as well as any frequency domain weights that are used to specify the performance objective. The objective of an optimal control problem is to synthesize a controller K that minimizes the closed-loop (e.g., H_2, H_∞, $\mathcal{L}_1$) norm from disturbances (d) to regulated variables (e), i.e., solve

$$\min_{\text{allowable } K} \|F_L(P, K)\|$$

where $F_L(P, K)$ denotes the system obtained by closing the controller K around the lower loop of P. The H_2, H_∞, and $\mathcal{L}_1$ optimal control problems refer to the choice of the specific norm $\|F_L(P, K)\|$ used to specify the performance. A generalization of the H_∞ performance objective is simply to require that the closed-loop map from $d \to e$ satisfy an IQC defined by a given multiplier Π, called the performance multiplier, Apkarian and Noll (2006). The H_2, H_∞ and $\mathcal{L}_1$ optimal control problems formulated as in Fig. 2 only involve signal uncertainty. In other words, these design problems do not explicitly account for the effects of model uncertainty.
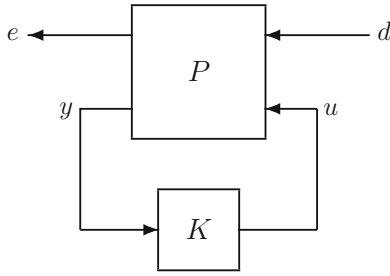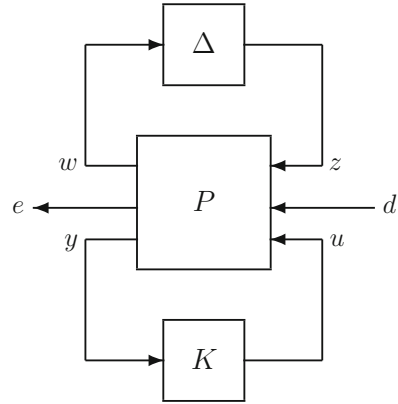
Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 2 Feedback interconnection for H_2, H_∞, and $\mathcal{L}_1$ optimal control

Robust Synthesis and Robustness Analysis Techniques and Tools, Fig. 3 Feedback interconnection for μ synthesis

Robust synthesis refers to control design that explicitly accounts for model uncertainty. It is usually formulated as a worst-case optimization, where the controller is chosen to minimize the worst-case effect of the signal and model uncertainty, loosely

$$\min_{\text{allowable } K} \max_{\text{allowable } d, \Delta} \|T(d, \Delta, K)\|$$

where d is a set of exogenous disturbances and Δ corresponds to the model uncertainty set. T represents the closed-loop relationship between d, Δ and the controller K. μ-synthesis is a specific technique developed to synthesize control algorithms which achieve robust performance, i.e., performance in the presence of signal and model uncertainty. The objective of μ-synthesis is to minimize over all stabilizing controllers K, the peak value of $\mu_\Delta (F_L(P, K))$ of the closed-loop transfer function defined by the interconnection in Fig. 3. P is the generalized plant model. The Δ block is the uncertain element from the set $\boldsymbol{\Delta}$, which parameterizes all of the assumed model uncertainty in the problem. The μ-synthesis optimization has high computational complexity (so-called NP-hard problem), though practical algorithms and software have been developed to design controllers using this control technique (Balas et al. 2013). Alternative robust synthesis approaches exist and often involve nonlinear optimization algorithms (Apkarian and Noll 2006). Drastic simplification regarding the models and uncertainty can be made resulting in problems that can be solved using LMI and

semidefinite programming techniques (Boyd and Barrat 1991; Boyd et al. 1994).

Computational Tools

The MATLAB Robust Control Toolbox is a commercially available software product that is part of the Mathworks control product line. It is tightly integrated with Control System Toolbox and Simulink products (Balas et al. 2013). The Robust Control Toolbox includes tools to analyze and design multi-input, multi-output control systems with uncertain elements. The primary building blocks, called uncertain elements or atoms, are uncertain real parameters and uncertain linear, time-invariant objects. These can be used to create coarse and simple or detailed and complex descriptions of model uncertainty. The uncertain object data structure eliminates the need to generate models of uncertainty and control analysis and design problem formulations, thereby allowing the practicing engineer to apply advanced robust control theory to their applications. Functions are available to analyze the robust stability, robust performance, and worst-case performance of uncertain multivariable system models using the structured singular value, μ. The Robust Control Toolbox also includes multivariable

control synthesis tools to compute controllers that optimize worst-case performance and identify worst-case parameter values.

The IQC-Beta Toolbox is a publicly available robust analysis toolbox based on the IQC framework (Jönsson et al. 2004). A wide range of robust stability and performance analysis tests are available for uncertain, nonlinear, and time-varying systems. IQC-Beta is written in MAT-LAB and works seamlessly with the Control System Toolbox objects and basic interconnection functions. The Users manual nicely complements the literature on IQCs. The Computer Aided Control System Design package in Scilab, an open source numerical computation software, includes functionality for robustness analysis and the synthesis of robust control algorithms for multivariable systems (http://www.scilab.org/).

Conclusions

Robust control analysis and synthesis software tools are widely available and have been extensively used by industry since the late 1980s. The availability of software tools for robustness analysis and synthesis played a major role in their wide and ubiquitous adoption in industry. They have been successfully applied to a variety of applications including aircraft flight control, launch vehicles, satellites, compact disk players, disk drives, backhoe excavators, nuclear power plants, helicopters, thin film extrusion, gas- and diesel-powered engines, missile autopilots, heating and ventilation systems, process control, and active suspension systems.

Cross-References

▶ LMI Approach to Robust Control
▶ Optimization Based Robust Control
▶ Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Bibliography

Apkarian P, Noll D (2006) IQC analysis and synthesis via nonsmooth optimization. Syst Control Lett 55:971–981

Balas GJ, Packard AK, Chiang RC, Safonov M (2013) MATLAB robust control toolbox. The Mathworks Inc.

Boyd S, Barrat C (1991) Linear controller design: limits of performance. Prentice Hall

Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia

Desoer C, Vidyasagar M (2008) Feedback systems: input–output properties. Classics in applied mathematics. SIAM, Philadelphia

Desoer C, Liu R, Murray J, Saeks R (1980) Feedback system design: a fractional representation approach to analysis and synthesis. IEEE Trans Autom Control 25(6):399–412

Doyle J (1978) Guaranteed margins for LQG regulators. IEEE Trans Autom Control 23(4): 756–757

Doyle J (1982) Analysis of feedback systems with structured uncertainties. IEE Proc Part D 129(6):242–251

Doyle J, Stein G (1981) Multivariable feedback design: concepts for a clasical/modern synthesis. IEEE Trans Autom Control 26(1):4–16

Doyle J, Glover K, Khargonekar P, Francis B (1989) State-space solutions to standard H_2 and H_∞ control problems. IEEE Trans Autom Control 34(8):831–847

Dullerud G, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York

Fan MKH, Tits AL, Doyle JC (1991) Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. IEEE Trans Autom Control 36(1):25–38

Francis B (1987) A course in $\mathcal{H}_\infty$ control theory. Lecture notes in control and information sciences, vol 88. Springer, Berlin/New York

Jönsson U, Kao CY, Megretski A, Rantzer A (2004) A guide to IQCβ: a MATLAB toolbox for robust stability and performance analysis

Megretski A, Rantzer A (1997) System analysis via integral quadratic constraints. IEEE Trans Autom Control 42(6):819–830

Packard A, Doyle J (1993) The complex structured singular value. Automatica 29(1):71–109

Safonov M (1982) Stability margins of diagonally perturbed multivariable feedback systems. IEE Proc Part D 129(6):251–256

Skogestad S, Postlethwaite I (2005) Multivariable feedback control: analysis and design. Wiley, Hoboken

Vidyasagar M (1985) Control system synthesis: a factorization approach. MIT, Cambridge

Willems JC (1971) Least squares stationary optimal control and the algebraic Riccati equation. IEEE Trans Autom Control 16:621–634

Zames G (1981) Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms,

and approximate inverses. IEEE Trans Autom Control 26(1):301–320

Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River

Robustness Analysis of Biological Models

Steffen Waldherr[1] and Frank Allgöwer[2]
[1]Institute for Automation Engineering, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
[2]Institute for Systems Theory and Automatic Control, University of Stuttgart, Stuttgart, Germany

Abstract

Robustness analysis is the process of checking whether a system's function is maintained despite perturbations. Robustness analysis of biological models is typically applied to differential equation models of biochemical reaction networks. While robustness is primarily a yes-or-no question, for many applications in biological models, it is also desired to compute a quantitative robustness measure. Such a measure is usually defined to be the maximum size of perturbations that the system can still tolerate. In addition, it is often of interest to specifically compute fragile perturbations, i.e., perturbations for which the system loses its function.

Keywords

Biochemical reaction networks; Fragile perturbations; Parametric uncertainty; Robustness measure; Structural uncertainty

Introduction

In biological systems analysis, robustness is the property that a system maintains its function in the face of internal or external perturbations (Kitano 2007). For a robustness analysis, one therefore needs to specify the system to be analyzed, the function that should be maintained, and the perturbation class.

The models to which robustness analysis is applied are mostly differential equation models of biochemical reaction networks. They are generally written as

$$\dot{x} = S v(x), \qquad (1)$$

where $x \in \mathbb{R}^n$ is the vector of intracellular concentrations; $S \in \mathbb{R}^{n \times m}$ is the stoichiometric matrix, containing the information how the individual network components participate in the reactions; and $v(x) \in \mathbb{R}^m$ is the reaction rate vector, in most cases a nonlinear function of the concentrations x.

The biological functions that are being studied by robustness analysis are very broad, pertaining to the wide range of biological functions implemented by biochemical reaction networks. Specific problems being considered are:

1. The occurrence of qualitative dynamical patterns such as sustained oscillations or multistability, where the system converges to one of multiple stable steady states depending on initial conditions or external stimuli (Eissing et al. 2005; Ma and Iglesias 2002).
2. The steady-state concentration value for a subset of the biochemical network's components (Shinar and Feinberg 2010; Steuer et al. 2011).
3. Quantitative measures derived from the network's dynamics, for example, the period of sustained oscillations (Stelling et al. 2004).

For the perturbation classes, two approaches can be distinguished. In parametric robustness analysis, a parametrized biological model is given, and the perturbation consists in varying the values of the parameters away from their nominal value. In structural robustness analysis, perturbations to the interaction structure of the network or the functional form of the reaction rate functions $v(x)$ are considered. Robustness analysis with these perturbation classes is presented in more detail below.

The perturbation class is also relevant for two applications of robustness analysis which

R

go beyond simply deciding whether a system is robust or not. First, it is often of interest to get a better quantification of robustness than a binary decision. Then, it is common to define a robustness measure, which usually quantifies how large perturbations can be without affecting the system's function (Ma and Iglesias 2002; Morohashi et al. 2002). Such a measure requires an appropriate definition of the perturbation size. With parametric perturbations, norms in parameter space are often useful (Ma and Iglesias 2002; Waldherr and Allgöwer 2011). With structural perturbations, the proximity of interaction functions in function space (Breindl et al. 2011) or the number of changes in the interaction structure can be evaluated.

Second, one often desires to compute specific non-robust perturbations, i.e., perturbations within the given class for which the system loses the considered functionality. There is a close relation between non-robust perturbations and the robustness measure, in that the norm of the smallest non-robust perturbation is equal to the robustness measure. Yet, it is often easier to compute a robustness measure than a non-robust perturbation. Especially algorithms that give a lower bound on the robustness measure will usually not provide a non-robust perturbation.

An illustration of the key characteristics in robustness analysis is shown in Fig. 1. This also illustrates that any norm-based robustness

measure depends on the nominal situation, where no perturbation is present.

When performing robustness analysis on a mathematical model of the considered system, the potential mismatch between model and system has to be kept in mind. By comparing the mathematical analysis results to experimental observations, robustness analysis methods are also useful for the validation or invalidation of biological network models (Bates and Cosentino 2011).

Robustness Analysis with Parametric Perturbations

Robustness analysis with parametric perturbations is applied to parametrized differential equation models of biochemical reaction networks, which are described by an equation of the form

$$\dot{x} = Sv(x, \mu), \qquad (2)$$

where $\mu \in \mathbb{R}^p$ is a vector of parameters. Such parameters may, for example, represent the total expression level of proteins involved in the reaction network, where usually a large variability due to the stochastic process of gene expression occurs.

This entry focuses on two specific system functionalities for robustness with respect to parametric perturbations, the qualitative dynamical behavior, and the steady-state level of a subset of the network's components. These are particularly relevant for biological models: the dynamical behavior often represents qualitative biological regulatory mechanisms, whereas the steady-state level of network components with a downstream regulatory effect is important for the stimulus-response relation of a biological network.

The Qualitative Dynamical Behavior

Considering the qualitative dynamical behavior, it is of interest to distinguish situations of a globally stable equilibrium point, multiple locally stable equilibrium points, or sustained oscillations due to a limit cycle or more complex attractors. Since changes in these dynamical patterns
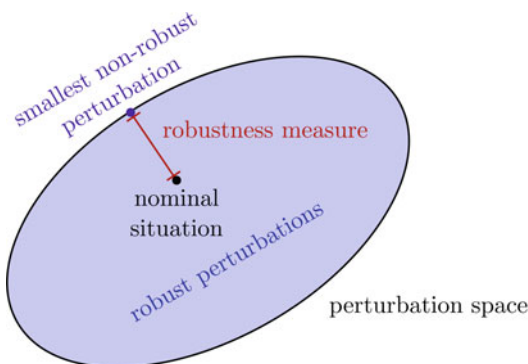
Robustness Analysis of Biological Models, Fig. 1 Illustration of key characteristics in robustness analysis

correspond to the occurrence of bifurcations in the model dynamics (2), this type of robustness analysis is closely related to bifurcation analysis. In the case of scalar, positive parameters μ, a corresponding robustness measure DOR has been defined by Ma and Iglesias (2002) as

$$DOR = 1 - \max\left\{ \frac{\check{\mu}}{\mu_0}, \frac{\mu_0}{\hat{\mu}}, 0 \right\}, \qquad (3)$$

where $\check{\mu}$ and $\hat{\mu}$ are the closest bifurcation points smaller and larger than μ, respectively. The robustness measure DOR is between 0 and 1 and indicates how much the parameter can be varied before reaching a bifurcation: for any multiplicative perturbation of less than $(1 - DOR)^{-1}$, no bifurcation will occur. A generalization to multiparametric models has been proposed in Waldherr and Allgöwer (2011): their robustness measure ϱ is defined as

$$\varrho = \sup\{\varrho \geq 1 \mid \text{no bifurcation occurs in the}$$
$$\text{hyperrectangle } [\varrho^{-1}\mu_0, \varrho\mu_0]\}. \qquad (4)$$

The measure ϱ directly gives the multiplicative parameter variation up to which no bifurcation occurs.

In general, the information required for a bifurcation-based robustness measure will only be available from a complete bifurcation analysis of the model. When restricting the types of bifurcations that are considered to bifurcations of equilibrium point, one can however check robustness by studying linear approximations at the system's equilibrium points. Since the reaction rates $v(x, \mu)$ are usually modeled as polynomial or rational functions, polynomial programming methods can be applied to compute a robustness measure (Waldherr and Allgöwer 2011) in this case.

The Steady-State Output Concentration

In biochemical network analysis, mostly linear outputs of the form

$$y = Cx, \qquad (5)$$

with $C \in \mathbb{R}^{q \times n}$ are considered. A common special case is that the rows of C are a subset of the rows of the identity matrix in $\mathbb{R}^n$, i.e.,

$$C = \left(e_i^{\mathrm{T}}\right)_{i \in \mathcal{I}_y}, \qquad (6)$$

and $\mathcal{I}_y \subset \{1, 2, \ldots, n\}$ is the index set defining the output concentrations.

A biochemical network has a robust steady-state output concentration, if the steady-state output $\bar{y}$ is independent of the parameters μ (Steuer et al. 2011). For a steady-state map $\bar{y} = h(\mu)$, this corresponds to the condition

$$h'(\mu) = 0. \qquad (7)$$

For the special case of an output given by (6), a sufficient and necessary condition for steady-state output robustness has been discovered by Steuer et al. (2011). The condition amounts to checking that a vector P, which describes the perturbation of the reaction rates under parameter variations, is in a subspace $\mathcal{I} = \operatorname{im} M + \ker S \operatorname{diag}(\alpha)$ for any α in the kernel of S, where M is a matrix composed of the normalized derivatives of the reaction rates with respect to the concentrations which do not appear in the output. A notable underlying assumption here is that the network's steady state does not undergo any local bifurcations within the considered parameter region, which directly relates back to the robustness analysis discussed in the previous section.

For the special case where parameters are the concentrations of conserved chemical species, a sufficient condition for steady-state output robustness has also been discovered by Shinar and Feinberg (2010). They propose the term *absolute concentration robustness* for this property. Here, the assumption that no local bifurcations occur within the considered parameter region is not required a priori but rather is also a consequence of the proposed condition.

Robustness Analysis with Structural Perturbations

Robustness analysis with parametric perturbations is based on the assumption that the reaction rate expressions are exact and that all perturbations are captured by parameter variations. This assumption can hardly be justified for many practical models, and an analysis with structural perturbations becomes necessary. Such analyses have discovered models which are very robust against parametric perturbations but nonrobust against structural perturbations (Jacobsen and Cedersund 2008).

The biological functions for which rigorous results on structural robustness are available are again related to the nonoccurrence of bifurcations in the model. For the restriction to local bifurcations of equilibria, linear systems theory offers efficient analysis tools for structural robustness.

In a first step, a structural perturbation of the network's interaction graph was suggested (Jacobsen and Cedersund 2008). This approach considers the network's Jacobian

$$A = S \frac{\partial v}{\partial x}(\bar{x}) \qquad (8)$$

evaluated at a steady state $\bar{x}$. The Jacobian is then perturbed to

$$\tilde{A} = \mathrm{diag}\, A + (A - \mathrm{diag}\, A)(I + \Delta), \qquad (9)$$

where $\mathrm{diag}\, A$ is the diagonal of A and Δ is a perturbation matrix, containing uncertain time-invariant linear systems as elements.

As an alternative approach, Waldherr et al. (2009) have suggested a structural perturbation of the reaction rate expressions. Thereby, the network's Jacobian is perturbed to

$$\tilde{A} = S \left(\frac{\partial v}{\partial x}(\bar{x}) + \Delta \right). \qquad (10)$$

In the case of real Δ, this perturbation simply corresponds to a change in the reaction rate slopes at steady state.

With both approaches, robustness analysis with structured singular values can be applied to test for changes in the local dynamics at the considered equilibrium point. This allows to evaluate a model's robustness against this type of structural perturbations and also yields non-robust perturbations.

Summary and Future Directions

Robustness analysis of biological models is well established in biological network theory. Mathematical methods rooted in systems and control are particularly beneficial for approaching this task.

While this entry focuses on models of biochemical reaction networks given by differential equations, the robustness analysis problem has also been studied in other model frameworks, for example, discrete dynamical models (Chaves et al. 2006). Yet, beyond simulation-based studies, robustness analysis is still an open problem in many practically relevant biological model classes. This concerns, for example, stochastic models or models on the cell population level.

In a similar manner, it will be important to extend the perturbation classes that are being considered and to include, for example, time-varying or other perturbations that are relevant for biological models. Concerning the biological function, most robustness analysis methods focus on the steady-state behavior. In the future, it will be of interest to also take, for example, the transient dynamics into account.

In linear systems theory, the concept of robust performance is well established. While efforts have been made to transfer that concept to biochemical networks (Doyle and Stelling 2005), it remains difficult to quantify performance of such networks, thus impeding the development of stringent robustness analysis tools. One of the reasons for this difficulty is certainly that biological performance is more naturally defined in the time domain than in the frequency domain, which narrows the conclusions that could be drawn from a direct application of classical robust performance analysis methods.

Cross-References

Bibliography

Bates D, Cosentino C (2011) Validation and invalidation of systems biology models using robustness analysis. IET Syst Biol 5(4):229–244

Breindl C, Waldherr S, Wittmann DM, Theis FJ, Allgöwer F (2011) Steady-state robustness of qualitative gene regulation networks. Int J Robust Nonlinear Control 21(15):1742–1758. doi:10.1002/rnc.1786, http://dx.doi.org/10.1002/rnc.1786

Chaves M, Sontag ED, Albert R (2006) Methods of robustness analysis for Boolean models of gene control networks. IEE Proc Syst Biol 153(4):154–167. doi:10.1049/ip-syb:20050079, http://dx.doi.org/10.1049/ip-syb:20050079

Doyle FJ, Stelling J (2005) Robust performance in biophysical networks. In: Proceedings of the 16th IFAC World Congress, Prague

Eissing T, Allgöwer F, Bullinger E (2005) Robustness properties of apoptosis models with respect to parameter variations and intrinsic noise. IEE Proc Syst Biol 152(4):221–228. doi:10.1049/ip-syb:20050046

Jacobsen EW, Cedersund G (2008) Structural robustness of biochemical network models—with application to the oscillatory metabolism of activated neutrophils. IET Syst Biol 2(1):39–47. http://link.aip.org/link/?SYB/2/39/1

Kitano H (2007) Towards a theory of biological robustness. Mol Syst Biol 3:137. doi:10.1038/msb4100179, http://dx.doi.org/10.1038/msb4100179

Ma L, Iglesias PA (2002) Quantifying robustness of biochemical network models. BMC Bioinform 3:38

Morohashi M, Winn AE, Borisuk MT, Bolouri H, Doyle J, Kitano H (2002) Robustness as a measure of plausibility in models of biochemical networks. J Theor Biol 216(1):19–30. doi:10.1006/jtbi.2002.2537, http://dx.doi.org/10.1006/jtbi.2002.2537

Shinar G, Feinberg M (2010) Structural sources of robustness in biochemical reaction networks. Science 327(5971):1389–1391. doi:10.1126/science.1183372, http://dx.doi.org/10.1126/science.1183372

Stelling J, Gilles ED, Doyle III FJ (2004) Robustness properties of circadian clock architectures. Proc Natl Acad Sci 101(36):13210–13215. doi:10.1073/pnas.0401463101, http://dx.doi.org/10.1073/pnas.0401463101

Steuer R, Waldherr S, Sourjik V, Kollmann M (2011) Robust signal processing in living cells. PLoS Comput Biol 7(11):e1002218. http://dx.doi.org/10.1371%2Fjournal.pcbi.1002218

Waldherr S, Allgöwer F (2011) Robust stability and instability of biochemical networks with parametric uncertainty. Automatica 47:1139–1146. doi:10.1016/j.automatica.2011.01.012, http://dx.doi.org/10.1016/j.automatica.2011.01.012

Waldherr S, Allgöwer F, Jacobsen EW (2009) Kinetic perturbations as robustness analysis tool for biochemical reaction networks. In: Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, pp 4572–4577. doi:10.1109/CDC.2009.5400939, http://dx.doi.org/10.1109/CDC.2009.5400939

Robustness Issues in Quantum Control

Ian R. Petersen
School of Engineering and Information Technology, University of New South Wales, the Australian Defence Force Academy, Canberra, Australia

Abstract

Robust quantum control theory is concerned with the design of controllers for quantum systems taking into account uncertainty is the model of the system. The robust open-loop control of quantum systems is discussed in this entry. Also discussed is the robust stability analysis problem for quantum systems, and two forms of quantum small gain theorem are presented. In addition, the entry discusses the design of robust quantum feedback control systems.

Keywords

Ensemble controllability; H^∞ control; Minimax control; Quantum control; Robustness; Robust stability

R

Introduction

The control of systems whose dynamics are governed by the laws of quantum mechanics is the subject of quantum control theory. The topic of quantum control theory is covered in the companion article Petersen (2014). As in the case of classical control theory, the models used in quantum control are often subject to uncertainties. This motivates the study of robust quantum control, in which the quantum systems to be controlled are modeled as uncertain quantum systems, e.g., see Mabuchi and Khaneja (2005). A related problem is the problem of robust estimation and filtering for uncertain quantum systems, e.g., see Yamamoto and Bouten (2009). The issue of robust stability is particularly important in the case of quantum feedback control since in this case, there is always the possibility of instability. An important area of quantum control theory is open-loop quantum control; see Petersen (2014). Since uncertainties arise in the quantum system models being considered, the robustness of open-loop quantum control systems is also important, e.g., see Li and Khaneja (2009), Rabitz (2002), and Owrutsky and Khaneja (2012).

This entry surveys some of the important research results on robust quantum control which have arisen in various application areas. These include some recent results on robust open-loop control of quantum systems; see Zhang and Rabitz (1994). Also considered are some recent results on robust stability analysis results for uncertain quantum systems, which amount to quantum versions of the classical small gain theorem; see Petersen et al. (2012). Finally, the entry looks at robust quantum feedback controller design; see James et al. (2008) and Dong et al. (2009).

Robust Open-Loop Control of Quantum Systems

In the robust open-loop control of quantum systems, the quantum system is modeled in the Schrödinger picture. The models can be given either in terms of the Schrödinger equation for the system state $|\psi(t)\rangle$:

$$i\frac{\partial}{\partial t}|\psi(t)\rangle = \left[H_0 + \sum_{k=1}^{m} u_k(t)H_k\right]|\psi(t)\rangle \quad (1)$$

or the master equation for the system density operator ρ:

$$\dot{\rho}(t) = -i\left[\left(H_0 + \sum_{k=1}^{m} u_k(t)H_k\right), \rho(t)\right] \quad (2)$$

e.g., see Petersen (2014). In these equations, H_0 is the free Hamiltonian of the system and H_k are corresponding control Hamiltonians. In the robust open-loop control of quantum systems, these quantities are assumed to be uncertain and the control law $u_k(t)$ is to be designed to guarantee an adequate level of performance for all possible values of the uncertainties. Here, performance is measured in terms of the fidelity between the actual final state or density matrix of the system and the desired final state or density matrix, e.g., see Nielsen and Chuang (2000).

In the minimax optimal control approach to robust open-loop control of quantum systems, the uncertainties in the Hamiltonian are represented in terms of a vector quantity w which is subject to constraints. Then, the robust control problem is the minimax optimal control problem

$$\min_u \max_w J(u, w)$$

where $J(u, w)$ is a suitable cost function, and the problem is subject to the constraints defined by the system dynamics (1) and the constraints on the uncertainty w; see Zhang and Rabitz (1994). Some standard numerical procedures have been proposed to solve this minimax optimal control problem with applications in chemical physics; see Zhang and Rabitz (1994).

Related to the robust open-loop control of quantum systems is the control of inhomogeneous quantum ensembles. In this problem, the same control signal $u_k(t)$ is applied to a large number of quantum particles in an ensemble. Also, the Hamiltonians corresponding

to individual particles may have different parameter values, and so this problem is equivalent to a robust open-loop quantum control problem, e.g., see Li and Khaneja (2006). In studying this problem, the issue of controllability has been considered (see, e.g., Li and Khaneja 2009) as in the standard open-loop quantum control problem; see Petersen (2014). Also, numerical methods have been proposed for constructing an optimal control law for inhomogeneous ensembles, e.g., see Ruths and Li (2012) and Owrutsky and Khaneja (2012). This approach has arisen in applications to chemical physics.

Robustness Analysis for Uncertain Quantum Systems

The problem of robust stability analysis for uncertain quantum systems was considered in the paper D'Helon and James (2006) which was concerned with the feedback interconnection of two quantum optical systems as shown in Fig. 1. In this interconnection, each of the quantum systems is linear quantum optical systems described in the Heisenberg picture by linear quantum stochastic differential equations (QSDEs) of the form

$$
\begin{aligned}
\mathrm{d}x(t) &= Ax(t)\mathrm{d}t + B\mathrm{d}u(t); \\
\mathrm{d}y(t) &= Cx(t)\mathrm{d}t + D\mathrm{d}u(t); \quad (3)
\end{aligned}
$$
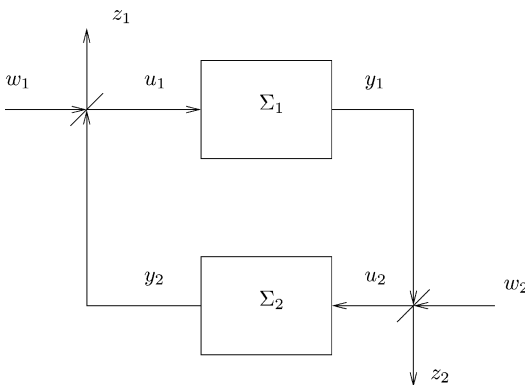
Robustness Issues in Quantum Control, Fig. 1 Feedback interconnection of two quantum optical systems

see James et al. (2008) and Petersen (2014) for more details on this class of quantum system models. Here, $x(t)$ are vector system variables which are operators on the underlying Hilbert space of the system. Also, the input and output fields are decomposed as $\mathrm{d}u(t) = \beta_u(t)\mathrm{d}t + \mathrm{d}\tilde{u}(t)$ and $\mathrm{d}y(t) = \beta_y(t)\mathrm{d}t + \mathrm{d}\tilde{y}(t)$ where $\beta_u(t)$, $\beta_y(t)$ denote the signal parts of the quantities $\mathrm{d}u(t)$, $\mathrm{d}y(t)$, respectively. Furthermore, $\mathrm{d}\tilde{u}(t)$, $\mathrm{d}\tilde{y}(t)$ denote the noise parts of the quantities $\mathrm{d}u(t)$, $\mathrm{d}y(t)$, respectively, e.g., see James et al. (2008). Such a system is stable and has a finite gain $g > 0$ if there exist constants $\mu > 0$ and $\lambda > 0$ such that

$$
\begin{aligned}
\int_0^t \langle \|\beta_y(\tau)\|^2 \rangle \mathrm{d}t &\leq \mu + \lambda t \\
&+ \int_0^t \langle \|\beta_u(\tau)\|^2 \rangle \mathrm{d}t \quad \forall t > 0;
\end{aligned}
$$

e.g., see D'Helon and James (2006) and James and Gough (2010). Here, $\langle \cdot \rangle$ denotes quantum expectation.

The two quantum optical systems shown in Fig. 1 are interconnected via beam splitters which are described by equations

$$
u_1 = \epsilon_1 w_1 - \sqrt{1 - \epsilon_1^2}\, y_2; \quad z_1 = \sqrt{1 - \epsilon_1^2}\, w_1 + \epsilon_1 y_2;
$$

$$
u_2 = \epsilon_2 w_2 - \sqrt{1 - \epsilon_2^2}\, y_1; \quad z_2 = \sqrt{1 - \epsilon_2^2}\, w_2 + \epsilon_2 y_1
$$

where $\epsilon_1 \in (0, 1)$ and $\epsilon_2 \in (0, 1)$ are given constants. The quantum small gain theorem established in D'Helon and James (2006) shows that if each of the quantum systems in Fig. 1 is stable and has finite gains $g_1 > 0$ and $g_2 > 0$ respectively such that $\sqrt{1 - \epsilon_1^2}\sqrt{1 - \epsilon_2^2}\, g_1 g_2 < 1$, then the feedback interconnected system will also be stable and have a finite gain. This result can be thought of as a stability robustness result if the first quantum system is regarded as the nominal quantum system and the second quantum system is regarded as being the uncertain part of the system subject to the given finite gain constraint.

An alternative approach to the robust stability analysis of uncertain quantum systems considers an uncertain quantum system described using the (S, L, H) description (see Petersen (2014) and

Gough and James (2009) for more details on this class of quantum systems). Here, the system Hamiltonian is described in terms of vectors of annihilation and creation operators a and $a^{\#}$, respectively, as

$$H = \frac{1}{2} \begin{bmatrix} a^{\dagger} & a^{T} \end{bmatrix} M \begin{bmatrix} a \\ a^{\#} \end{bmatrix} + \frac{1}{2} \tilde{\xi}^{\dagger} \Delta \tilde{\xi}$$

where M is a known complex Hermitian matrix describing the nominal Hamiltonian, Δ is a complex Hermitian uncertainty matrix subject to the norm bound $\|\Delta\| \leq \frac{2}{\gamma}$, and $\tilde{\xi} = E \begin{bmatrix} a \\ a^{\#} \end{bmatrix}$. Also, E is a known complex matrix describing the uncertainty structure. Furthermore, it is assumed that $S = I$ and the coupling operator vector L is such that $\begin{bmatrix} L \\ L^{\#} \end{bmatrix} = N \begin{bmatrix} a \\ a^{\#} \end{bmatrix}$ where N is a known complex matrix. This uncertain quantum system is robustly mean square stable if the H^{∞} norm bound condition

$$\left\| E \left(sI + iJM + \frac{1}{2} JN^{\dagger} JN \right)^{-1} JE^{\dagger} \right\|_{\infty} < \frac{\gamma}{2}$$

is satisfied where $J = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}$; see Petersen et al. (2012).

Robust Feedback Control of Quantum Systems

Schrödinger Picture Approaches to Robust Measurement-Based Quantum Feedback Control

A number of results have appeared which use Schrödinger picture models (see Petersen 2014) in robust measurement-based quantum feedback control. These results are based on uncertain quantum system models of the form (1) or (2) and extend the results mentioned above by allowing for measurements of the quantum system in order to achieve improved robustness against uncertainties in the system Hamiltonian. For example, consider a quantum system of the form (1) with uncertainties in the system Hamiltonian. Then a

measurement feedback robust control scheme can be constructed which involves periodic projective measurements on the system. In a projective measurement of the quantum system (1), the state $|\psi(t)\rangle$ collapses to an eigenstate of H_0 corresponding to the measurement outcome obtained. The sliding mode control algorithm uses open-loop time optimal control (see Petersen 2014) to steer the state of the system back to a specified eigenstate of the system whenever a measurement is obtained which does not correspond to this desired eigenstate; see Dong and Petersen (2009). This desired eigenstate is referred to as the sliding mode domain, and the state of the system is guaranteed to stay within the sliding mode domain with a specified probability provided that the measurement sampling period in the proposed feedback control algorithm is chosen to be sufficiently fast; see Dong and Petersen (2009). In the case of two-level quantum systems, this sliding mode control approach is implemented using a Lyapunov method for open-loop quantum control to steer the system back to the sliding mode domain; see Petersen (2014) and Dong and Petersen (2012). In all of these cases, robustness is ensured by including uncertainty in the underlying quantum system models and then taking this into account in the design of the control laws and sampling period.

Another approach to the measurement-based robust quantum feedback control problem involves an extension of the robust open-loop control results considered in section "Robust Open-Loop Control of Quantum Systems." In this approach, robust open-loop control results are extended to solve the problem of stabilization of an ensemble of quantum particles; see Beauchard et al. (2012).

Heisenberg Picture Approaches to Robust Quantum Feedback Control

Consider a quantum linear system modeled in the Heisenberg picture by quantum stochastic differential equations (QSDEs) as follows:

$$\begin{aligned} dx(t) &= Ax(t)dt + Bdw(t); \\ dy(t) &= Cx(t)dt + Ddw(t); \end{aligned} \qquad (4)$$

see Petersen (2014) for details on this class of quantum system models which arises in the area of quantum optics. In the robust quantum feedback control problem, the matrices A, B, C, D may be uncertain and a feedback controller can be designed using the quantum H^∞ control approach to ensure that the resulting closed-loop system is robustly stable; see James et al. (2008). In the case of measurement-based feedback control, the controller is a classical system described by linear stochastic differential equations of the form

$$dx_K(t) = A_K x_k(t)dt + B_K dy(t)$$
$$\beta_u(t)dt = C_K x_k(t)dt; \qquad (5)$$

see Petersen (2014). In the case of coherent feedback control, the controller is another quantum linear system described by QSDEs of the form

$$dx_K(t) = A_K x_k(t)dt + B_K dy(t) + \bar{B}_K d\bar{w}_K(t)$$
$$dy_K(t) = C_K x_k(t)dt + \bar{D}_K d\bar{w}_K(t); \qquad (6)$$

see Petersen (2014).

In this approach to robust quantum feedback control, the uncertainty in the quantum system being controlled is represented by uncertainty in the matrix A as $A = \tilde{A} + \tilde{B}\Delta\tilde{C}$ where Δ is a constant but unknown uncertain matrix satisfying the bound $\Delta^T \Delta \leq I$. The controller, which may be either a classical controller or a coherent controller, is designed using the quantum H^∞ approach. Then the resulting closed-loop system will be robustly stable; see James et al. (2008). Similarly, in the case of uncertainty in the plant Hamiltonian matrix such as considered in section "Robustness Analysis for Uncertain Quantum Systems" or uncertainty in the form of an uncertain subsystem connected optically to the plant in feedback, also as considered in section "Robustness Analysis for Uncertain Quantum Systems," then the quantum H^∞ approach combined with the robust stability analysis results of section "Robustness Analysis for Uncertain Quantum Systems" shows that the quantum H^∞ method can also be used to design robustly stabilizing controllers in these cases.

Summary and Future Directions

To date there have been only a few papers published in the general area of robust quantum control. The results which were considered in this entry covered open-loop and feedback quantum control problems along with stability robustness analysis problems. A common theme in the results which were considered is that they were based on uncertain quantum mechanical models. It is expected that future research in this area will intensify as the use of feedback control becomes more prevalent in areas of experimental quantum technology.

Cross-References

▶ Control of Quantum Systems
▶ H-Infinity Control
▶ LMI Approach to Robust Control
▶ Optimization Based Robust Control

Bibliography

Beauchard K, da Silva PSP, Rouchon P (2012) Stabilization for an ensemble of half-spin systems. Automatica 48(1):68–76

D'Helon C, James M (2006) Stability, gain, and robustness in quantum feedback networks. Phys Rev A 73:053803

Dong D, Petersen IR (2009) Sliding mode control of quantum systems. New J Phys 11:105033

Dong D, Petersen IR (2012) Sliding mode control of two-level quantum systems. Automatica 48(5):725–735

Dong D, Lam J, Petersen IR (2009) Robust incoherent control of qubit systems via switching and optimization. Int J Control 83(1):206–217

Gough J, James MR (2009) The series product and its application to quantum feedforward and feedback networks. IEEE Trans Autom Control 54(11):2530–2544

James M, Gough J (2010) Quantum dissipative systems and feedback control design by interconnection. IEEE Trans Autom Control 55(8):1806–1821

James MR, Nurdin HI, Petersen IR (2008) H^∞ control of linear quantum stochastic systems. IEEE Trans Autom Control 53(8):1787–1803

Li J-S, Khaneja N (2006) Control of inhomogeneous quantum ensembles. Phys Rev A 73:030302

Li J-S, Khaneja N (2009) Ensemble control of Bloch equations. IEEE Trans Autom Control 54(3):528–536

R

Mabuchi H, Khaneja N (2005) Principles and applications of control in quantum systems. Int J Robust Nonlin Control 15:647–667

Nielsen M, Chuang I (2000) Quantum computation and quantum information. Cambridge University Press, Cambridge

Owrutsky P, Khaneja N (2012) Control of inhomogeneous ensembles on the Bloch sphere. Phys Rev A 86:022315

Petersen IR (2014) Quantum control. In: Samad T, Baillieul J (eds) Encyclopedia of systems and control. Springer, Heidelberg/Germany

Petersen IR, Ugrinovskii V, James MR (2012) Robust stability of uncertain linear quantum systems. Philos Trans R Soc A 370(1979):5354–5363

Rabitz H (2002) Optimal control of quantum systems: origins of inherent robustness to control field fluctuations. Phys Rev A 66:063405

Ruths J, Li J-S (2012) Optimal control of inhomogeneous ensembles. IEEE Trans Autom Control 57(8):2021–2032

Yamamoto N, Bouten L (2009) Quantum risk-sensitive estimation and robustness. IEEE Trans Autom Control 54(1):92–107

Zhang H, Rabitz H (1994) Robust optimal control of quantum molecular systems in the presence of disturbances and uncertainties. Phys Rev A 49:2241–2254

RTO

Run-to-Run Control in Semiconductor Manufacturing

James Moyne
Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA

Abstract

Run-to-run (R2R) control is a form of adaptive model-based process control that can be tailored to environments where the process is discrete, dynamic, and highly unobservable; this is characteristic of processes in the semiconductor manufacturing industry. It generally has, at its roots, a rather straightforward approach to adaptive model-based control such as a first-order linear plant model with moving average weighting applied to adapt the (zeroth-order) constant term in the model. Most of the complexity of R2R control science lies and will continue to lie in extensions to support practical application of R2R control in semiconductor manufacturing facilities of the future; these extensions include support for weighting and bounding of parameters, run-time modeling of a large number of disturbance types, and incorporating prediction information such as virtual metrology and yield prediction into the control solution.

Keywords

Adaptive control; Advanced process control (APC); EWMA control; Feed-forward and feedback control; Model-based control; R2R control; Run-to-run control; Single-threaded control; Virtual metrology; Wafer-to-wafer control; Yield prediction

Introduction

The semiconductor manufacturing industry involves the processing of semiconductor "wafers" using a variety of physical and chemical processes to produce dies or "chips" that contain a number of nanometer size features organized in layers. As feature sizes shrink, the industry must innovate to maintain acceptable product yield and throughput. One effective dimension of innovation that has been utilized since the early 1990s is model-based process control. The use of this technology in semiconductor manufacturing has been largely industry specific due to unique industry requirements and been given the name "run-to-run (R2R) control."

R2R control is defined as "...a form of discrete process and machine control in which the product recipe with respect to a particular machine process is modified ex situ, i.e., between
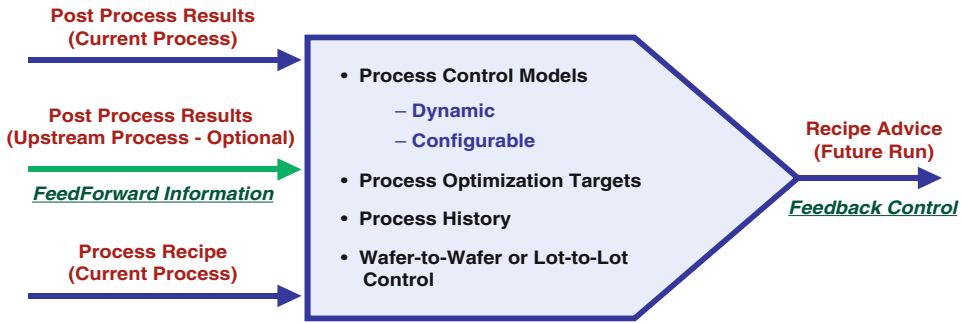
Run-to-Run Control in Semiconductor Manufacturing, Fig. 1 Input/output structure of a typical R2R control solution

machine 'runs,' so as to minimize process drift, shift, and variability" (Moyne et al. 2000). (The "recipe" is the group of process settings for a process or process step, e.g., temperature, flow, and pressure.) The term "R2R control" was coined in the early 1990s in the semiconductor industry as the industry struggled to come up with mechanisms to keep critical semiconductor manufacturing processes such as chemical vapor deposition (CVD), chemical mechanical polishing (CMP), and reactive ion etching (RIE) under control. The processes are highly unobservable and are subjected to a number of disturbances. However, many of these disturbances can be modeled or tracked as they create measurable shifts in the process (e.g., after a maintenance operation) or gradual drifts in the process (e.g., chamber wall "seasoning" of an etch process over time, resulting in polymer buildup on chamber walls, causes changes to the operational effectiveness of the tool). Process and product quality is generally assessed through metrology measurements made ex situ, i.e., after the process is complete; examples of post-process metrology parameters are wafer average deposited or removed film thickness and film uniformity. R2R control generally uses statistically developed models of tool process operation updated or "tuned" with process metrology feedback information on a "run-to-run" basis to keep the process under control and process quality high, in the face of these process drifts and shifts, as shown in Fig. 1. Note that the granularity of control could be wafer-to-wafer, or batch-to-batch ("lot-to-lot"), etc.

Run-to-Run Control Approach

Because the processes are highly unobservable and dynamic, rather simple model forms are usually employed with filtering techniques used to track process shift and drift. The most commonly utilized R2R controller in the industry is the exponentially weighted moving average (EWMA) controller. The algorithm uses a linear model with an additional constant term. (Equations will use the following notation: arrays of vectors will be capitals, vectors will be lower case, and indexing within a vector or matrix will be lower case with subscripts. In addition, the special subscript "t" will be reserved for time or run number information.)

$$Y = Ax + c \qquad (1)$$

where:

y = System output,
x = Input (Recipe),
A = Slope coefficients for equation,
c = Constant term for linear model.

Each output represents a target of control (usually measured by pre- and post-process metrology tools), and each input represents an adjustable parameter in the recipe.

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots a_{1m}x_m + c_1$$

$$\cdots$$

$$y_n = a_{n1}x_1 + a_{n2}x_2 + \ldots a_{nm}x_m + c_n \qquad (2)$$

R

The models are generally developed by executing a design of experiments (DOE), where the process area is explored with respect to the allowed variation of the process inputs by processing wafers with various input settings (see, e.g., Box and Draper 1987). Statistical packages are then used to determine the base model of the form described in (1) at the normal process operating point. As the processes are dynamic, the base model is updated on a "run-to-run" basis to compensate for model error. The algorithm operates under the assumption that the underlying process is locally approximated by a first-order linear polynomial model in the form of equation (1) and that this polynomial model can be maintained near a local optimal point solely by updating the constant term "c."

The control process involves updating the model and then using that model to compute a recipe update. The model is updated by first comparing the actual process output, Y_t, to the model-predicted process output, AX_t. Using an EWMA filtering technique as an example, the constant term, c_t can be updated as follows:

$$c_t = \alpha(y_t - Ax_t) + (1 - \alpha)c_{t-1} \qquad (3)$$

where α is a weighting factor between 0 and 1, often called a "forgetting factor." Note that because of the additive nature of the EWMA series, the C_t calculation only requires knowledge (and storage) of the previous run measurements; this, combined with its relative simplicity, led to the widespread adoption of EWMA as the R2R controller filter of choice in this industry during the 1990s and early 2000s.

Once the model is updated, the process recipe is calculated. Since there are generally more inputs that can be tuned than outputs measured, the process is underdetermined and there is an infinite solution space. Approaches such as Lagrange multipliers are used to determine the solution that is closest to the previous solution (Moyne et al. 2000).

Many extensions and alternatives to this basic approach have been developed and deployed over the past 10 years. These include (1) the replacement of EWMA filtering with other approaches such as the more general Kalman filtering, (2) explicitly modeling drift (termed "predictor corrector"), (3) modeling updates to first-order terms (in the "A" matrix), and (4) leveraging phenomenological models that capture process knowledge in equation forms, customized and tuned with statistical data. Perhaps the most important extensions to the basic approach involve addressing the practical issues associated with control systems application in this area. For example, providing capabilities for addressing bounding, weighting, and granularity (e.g., integer) of input and output settings often requires much more programming effort than supporting the core algorithm (Moyne et al. 2000).

Current Status and Future Extensions

Over the past 10 years, R2R control has evolved from a value-added capability applied to a few processes, to a required component to achieve cost and productivity competitiveness in most processes in the semiconductor manufacturing industries (ITRS 2014). As part of this evolution, a number of common trends in the R2R control space have emerged:

Support for fab-wide reusable and reconfigurable solutions for R2R control: As the benefits of R2R control were proven across multiple processes in semiconductor fabrication facilities, the focus turned to reusable and reconfigurable integrated "fab-wide" solutions for R2R control. The event-based capabilities described in Chapter 9 of Moyne et al. (2000) were leveraged to provide these solutions as they allow for integration and configuration of R2R control solutions to the particular application environment. This event-based approach has also been used to integrate R2R control with other capabilities such as fault detection and classification (FDC), work scheduling, and "virtual metrology" (see below), to provide another level of benefits towards improved product yield and throughput (Khan et al. 2007; Moyne 2004, 2009).

Movement to more granular control: The evermore stringent requirements on product quality are being addressed in large part by a movement from batch-level control (often called "lot-based control" in this domain), to wafer-level control (usually called "wafer-to-wafer" (W2W) control), to within-wafer (WIW) control. Although the granularity has changed, the basic approach to control has not. It is important to note that the improvement in quality associated with this trend results mostly from the use of pre-(process) metrology to reject incoming product disturbances, rather than post metrology to address the dynamics of the plant model (ITRS 2014; Moyne et al. 2000).

Support for control across multiple recipes using "single-threaded control": Semiconductor manufacturing process control systems are characterized by a number of disturbance types that usually can be modeled as independent from the base process model and from each other. Perhaps the most common type of disturbance that is addressed is recipe or product change. When there is a change in product and related product recipe, a single-process model must be adjusted to capture this disturbance while maintaining knowledge of process drift and/or shift. Oftentimes this process disturbance can be modeled as a shift to the overall process. Thus, the process model of equation (1) can be adjusted to the following:

$$Y = Ax + c_1 + c_2 + c_3 + \ldots + c_n \quad (4)$$

where:

$c_1, c_2, \ldots c_{n-1}$ = constant terms associated with modeled disturbances such as product

c_n = constant term associated with process dynamics (drift and shift)

$c_1 + c_2 + \ldots c_n = c$ in Eq. (1)

Approaches have been devised for the assessment of c_i associated with a particular disturbance type (Edgar et al. 2004; Zou 2013); the result is that a single-control model can be used across multiple product recipes and other disturbance types.

Enhancing R2R control with "virtual metrology": Ex-situ metrology plays a crucial role in semiconductor manufacturing as it is often the only source of product quality data before and after a process. However, given its high capital equipment cost and cycle time impact on critical processes, optimizing metrology by minimizing wasteful use and optimizing measurement value is important. Virtual metrology (VM) is a new technology rapidly gaining acceptance in the marketplace as an efficient and cost-effective way to optimize and augment metrology value. VM is a modeling and metrology prediction solution whereby process and product data, such as in situ fault detection (FD) information and upstream metrology information, is correlated to post-process metrology data. This same data can then be used to predict metrology information when conventional metrology data is not available (Cheng et al. 2011; Khan et al. 2007).

One of the uses of VM that is expected to become prominent over the next decade is in support of enhanced R2R control. As shown in Fig. 2, fault detection (FD) summary information

Run-to-Run Control in Semiconductor Manufacturing, Fig. 2 Virtual metrology enhanced R2R control
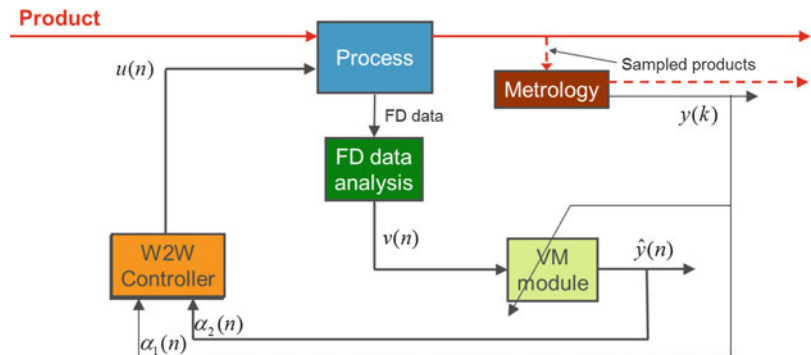
is used along with adaptive VM modeling to predict metrology information. The VM predictions are then used to fill in the measurement gaps in feed-forward and feedback control thus enabling wafer-to-wafer or even within-wafer control. One of the research challenges is to optimally tune the control to best utilize both the real and predicted metrology information. This requires that VM data contain information on predicted measurement data quality (Khan et al. 2007).

$u(n)$ Tunable process inputs

$v(n)$ FD summary information

$y(k)$ Metrology measurement data for measured wafers

$\hat{y}(n)$ Predicted metrology measurements for all wafers

$\alpha_1(n)$ Feedback filter coefficient for feedback of measured data

$\alpha_2(n)$ Feedback filter coefficient for feedback of predicted data

Movement towards interprocess and eventually fab-wide control: The generally accepted vision of the future of advanced process control (APC) in general is a fabrication-wide fully integrated solution that incorporates all of the APC capabilities (R2R control, FDC, fault prediction, and statistical process control) as well as predictive capabilities such as predictive scheduling, predictive maintenance, virtual metrology, and predictive yield (ITRS 2014). Opportunities for research and development exist with the integration of these technologies, especially as the powers of the predictive domain are tapped. For example, it is expected that R2R control will eventually incorporate predicted yield as a target with feedback to multiple coordinated process controllers (Moyne and Schulze 2010). Thus, the future of research in R2R control, while evolving, should remain strong in the coming years.

Summary and Future Directions

R2R control is a form of adaptive model-based process control that is tailored to environments where the process is discrete, dynamic, and highly unobservable; this is characteristic of processes in the semiconductor manufacturing industry. R2R control has evolved from a strictly research effort in the early 1990s to a required facility-wide capability in all of semiconductor manufacturing. It generally has, at its roots, a rather straightforward approach to adaptive model-based control. Most of the complexity of R2R control science lies and will continue to lie in extensions to support practical application of R2R control in semiconductor manufacturing facilities of the future.

The science of R2R control will continue to expand as the academic and industry communities look to incorporating capabilities that will allow R2R control to continue to be an integral part of the fabrication facility of the future. One key research direction over the next decade is the development of approaches for incorporating virtual metrology and yield prediction into control solutions. Other focus areas will likely include hybrids of R2R control and continuous process control, learning mechanisms for single-threaded control in "high-mix" environments where there are a large number of disturbances that should be modeled, phenomenological R2R control models, and model libraries that combine stochastic information with process physics and chemistry knowledge, control solutions that are more directly optimized to financial parameters such as yield and throughput, and R2R control solutions that incorporate other analysis capabilities, such as FDC, either algorithmically or via event-based control rule approaches. Each of these topics provides significant opportunity for research as well as benefit in application to semiconductor manufacturing facilities.

Cross-References

▸ Adaptive Control, Overview
▸ Controllability and Observability
▸ Event-Triggered and Self-Triggered Control
▸ Experiment Design and Identification for Control
▸ Fault Detection and Diagnosis
▸ Kalman Filters
▸ Moving Horizon Estimation
▸ Nominal Model-Predictive Control

▶ Robust Model-Predictive Control
▶ Stochastic Model Predictive Control

Bibliography

Advanced Process Control (APC) Conference Proceedings (2000–2014), titles available at http://www.apcconference.com

Box GEP, Draper NR (1987) Empirical model-building and response surfaces. Wiley, New York

Cheng F-T et al. (2011) Benefit model of virtual metrology and integrating AVM Into MES. IEEE Trans Semicond Manuf 24(2):261–272

Edgar TF, Firth SK, Bode C (2004) Multi-product run-to-run control for high-mix fabs. (session keynote) *AEC/APC Asia*, Hsinchu, (2004), (available at http://140.113.156.45/files/reference/2nd%20AEC-APC%20Symposium%20Asia/APCAEC/presentations/session_KeynoteInvited/Edgar_Thomas.pdf)

International Technology Roadmap for Semiconductors (ITRS), 2014 Edition, Semiconductor Industry Association. Available at http://www.itrs.net. (See especially "Factory Integration" chapter)

Khan A, Moyne J, Tilbury D (2007) Fab-wide control utilizing virtual metrology (invited). IEEE Trans Semicond Manuf-Spec Issue on Adv Process Control 20(7):364–375

Moyne J (2004) The evolution of APC: the move to total factory control (invited). Solid State Technol 47(9):47–52

Moyne J (2009) A blueprint for enterprise-wide deployment of advanced process control. Solid State Technol 52(7):35–37

Moyne J, Del Castillo E, Hurwitz A (2000) Run-to-run control in semiconductor manufacturing. CRC, Boca Raton

Moyne J, Schulze B (2010) Yield management enhanced advanced process control system (YMeAPC): part I, description and case study of feedback for optimized multi-process control. IEEE Trans Semicond Manuf Spec Issue Adv Process Control 23(2): 221–235

Zou J (2013) Method and system for estimating context offsets for run-to-run control in a semiconductor fabrication facility. United States Patent, Patent Number US 8,355,810 B2 (Filed, Jan 2010; Issued, Jan 2013)

R

S

Sampled-Data H-Infinity Optimization

Tongwen Chen
Department of Electrical and Computer
Engineering, University of Alberta, Edmonton,
AB, Canada

Abstract

$\mathcal{H}_\infty$ optimization is central in robust control. When controllers are implemented by computers, sampled-data control systems arise. Designing $\mathcal{H}_\infty$-optimal controllers in purely continuous time or in purely discrete time is standard in robust control; in this entry, we discuss the process of sampled-data optimization, namely, designing digital controllers based on a continuous-time $\mathcal{H}_\infty$ performance measure.

Keywords

Computer control; $\mathcal{H}_\infty$ discretization; Robust control; Sampled-data systems

Introduction

Robust control deals mainly with controller design against uncertainties in system modeling and disturbances. The central tool used is $\mathcal{H}_\infty$ optimization.

In continuous time, consider the standard setup in Fig. 1, where G is the generalized plant and K is the controller; G has two inputs (w, the exogenous input, and u, the control input) and two outputs (z, the output to be controlled, and y, the measured output); K processes y to generate u. The $\mathcal{H}_\infty$-optimal control problem is to design K to stabilize G and minimize the $\mathcal{H}_\infty$ norm of the closed-loop system in Fig. 1 from w to z, denoted T_{zw}. When both G and K are continuous-time, linear time-invariant (LTI), the $\mathcal{H}_\infty$ norm, $\|T_{zw}\|$, relates to the frequency response matrix $\widehat{T}_{zw}(j\omega)$ as follows:

$$\|T_{zw}\| = \sup_\omega \bar{\sigma}\left[\widehat{T}_{zw}(j\omega)\right],$$

where $\bar{\sigma}$ indicates the maximum singular value. This $\mathcal{H}_\infty$-optimal control problem in the LTI case is solvable by many techniques, e.g., Riccati equations and linear matrix inequalities – see robust control textbooks by Zhou et al. (1996) and Dullerud and Paganini (2000).

Sampled-Data Control

When controllers are implemented by digital computers, periodic samplers and zero-order holds are used to model analog-to-digital and digital-to-analog conversion. Replacing K in Fig. 1 by sampler S (with period h), discrete-time controller K_d, and zero-order hold H (synchronized with S), we obtain a sampled-data
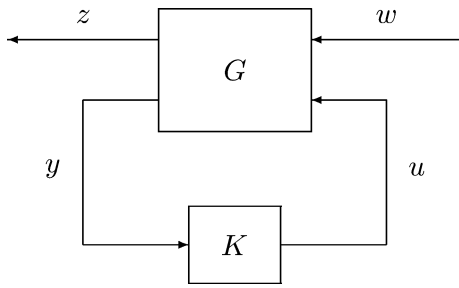
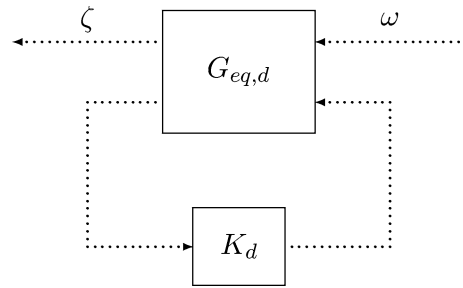**Sampled-Data H-Infinity Optimization, Fig. 1**
Standard control setup in continuous time
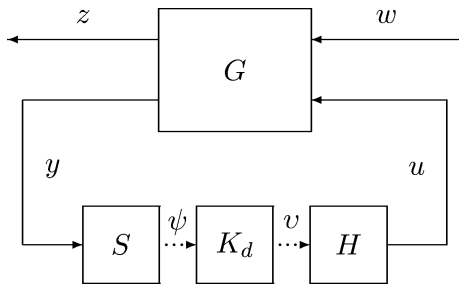
Sampled-Data H-Infinity Optimization, Fig. 2
Sampled-data control setup

Sampled-Data H-Infinity Optimization, Fig. 3 The
equivalent discrete-time system

control system shown in Fig. 2; here, S converts y into a discrete-time sequence ψ; K_d, a real-time algorithm in the computer, inputs ψ and computes another sequence υ, which is converted by H into u.

There are in general three approaches to design a digital controller K_d: design a continuous-time controller K and then implement digitally via approximation, discretize the plant and then design K_d in discrete time, and finally, design K_d directly based on continuous-time performance specifications (Chen and Francis 1995). The last approach is followed in the $\mathcal{H}_\infty$ optimization framework.

Sampled-Data $\mathcal{H}_\infty$ Discretization

The sampled-data $\mathcal{H}_\infty$ control problem is to design K_d *directly* to stabilize G in Fig. 2 and minimize $\|T_{zw}\|$. Notice that even if G is LTI in continuous time and K_d is LTI in discrete time, the closed-loop system T_{zw} is no longer LTI, due to the presence of S and H in the control loop;

in this case, the $\mathcal{H}_\infty$ norm is interpreted as the $\mathcal{L}_2$-induced norm:

$$\|T_{zw}\| = \sup\{\|z\|_2 : \|w\|_2 = 1\};$$

here, $\|\cdot\|_2$ represents the $\mathcal{L}_2$ norm on signals.

The sampled-data $\mathcal{H}_\infty$ control problem has been shown to be equivalent to a purely discrete-time $\mathcal{H}_\infty$ control problem (Kabamba and Hara 1993; Bamieh and Pearson 1992; Toivonen 1992); the process is known as sampled-data $\mathcal{H}_\infty$ discretization: for $\gamma > 0$, construct an LTI discrete-time system $G_{eq,d}$ connected to K_d as in Fig. 3; the two systems, T_{zw} in Fig. 2 and $T_{\zeta\omega} : \omega \mapsto \zeta$ in Fig. 3, are *equivalent* in that $\|T_{zw}\| < \gamma$ if $\|T_{\zeta\omega}\| < \gamma$, where the latter norm is ℓ_2-induced, and since $T_{\zeta\omega}$ is LTI in discrete time, it equals the $\mathcal{H}_\infty$ norm of the corresponding transfer function $\widehat{T}_{\zeta\omega}(z)$. Thus, pure discrete-time techniques are immediately applicable.

There are several ways to present this discretization. However, the computation is quite involved and hence is not given here; interested readers can find details in the papers by Kabamba and Hara (1993), Bamieh and Pearson (1992), and Toivonen (1992), or the book by Chen and Francis (1995). Note that the $\mathcal{H}_\infty$ discretization process is not quite *exact* in the sense that $G_{eq,d}$ depends on γ (Chen and Francis 1995).

Summary and Future Directions

In sampled-data $\mathcal{H}_\infty$ optimization, the key idea is to address the hybrid nature of the problem,

considering intersample behavior in formulation; the main tool is the so-called continuous lifting (Yamamoto 1994; Bamieh and Pearson 1992), making use of periodicity of sampled-data systems.

The ideas and tools developed in sampled-data control theory are still being used in emerging areas such as hybrid systems and networked control systems. For example, in event-triggered control systems, information exchange and control updating are not time driven but are done by certain event-triggering schemes, resulting in necessarily nonlinear and time-varying closed-loop dynamics; the analysis and synthesis issues in such systems are still challenging.

Cross-References

- ▶ H-Infinity Control
- ▶ LMI Approach to Robust Control
- ▶ Optimal Sampled-Data Control
- ▶ Optimization Based Robust Control

Recommended Reading

The continuous-time $\mathcal{H}_\infty$ control problem and its solutions are discussed extensively in several textbooks, e.g., Zhou et al. (1996) and Dullerud and Paganini (2000). The discrete-time $\mathcal{H}_\infty$ control problem was solved via the approach of Riccati equations in Iglesias and Glover (1991). The sampled-data $\mathcal{H}_\infty$ control problem was solved simultaneously with different methods in Kabamba and Hara (1993), Bamieh and Pearson (1992), and Toivonen (1992); details of the solution discussed here can be found in the book by Chen and Francis (1995).

Bibliography

Bamieh B, Pearson JB (1992) A general framework for linear periodic systems with application to $\mathcal{H}_\infty$ sampled-data control. IEEE Trans Autom Control 37:413–435

Chen T, Francis BA (1995) Optimal sampled-data control systems. Springer, London

Dullerud GE, Paganini F (2000) A course in robust control theory: a convex approach. Springer, New York

Iglesias P, Glover K (1991) State-space approach to discrete-time $\mathcal{H}_\infty$ control. Int J Control 54:1031–1073

Kabamba PT, Hara S (1993) Worst case analysis and design of sampled-data control systems. IEEE Trans Autom Control 38:1337–1357

Toivonen HT (1992) Sampled-data control of continuous-time systems with an $\mathcal{H}_\infty$ optimality criterion. Automatica 28:45–54

Yamamoto Y (1994) A function space approach to sampled data control systems and tracking problems. IEEE Trans Autom Control 39:703–713

Zhou K, Doyle J, Glover K (1996) Robust and optimal control. Prentice Hall, Upper Saddle River, New Jersey

---

Sampled-Data Systems

Panos J. Antsaklis[1] and H.L. Trentelman[2]
[1]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
[2]Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, Groningen, AV, The Netherlands

Abstract

For digital devices to interact with the physical world, an interface is needed that transforms the signals from analog to digital and vice versa. Ideal samplers and zero-order hold devices are incorporated to derive discrete-time models of continuous-time systems. State variable descriptions and transfer functions are used.

Keywords

Continuous-time approximations; Digital control; Discrete-time approximations; Quantization; Reconstruction; Sampled-data systems; Sampling

Introduction

Sampled-data systems are discrete-time models of continuous-time processes useful in the digital

S

control of continuous-time systems. A digital controller cannot communicate directly with a continuous system and an interface is needed.

Consider a continuous-time system having $u(t)$ as its input and $y(t)$ as its output.

A/D Converter: The continuous-time signal $y(t)$ is converted into a discrete-time signal $\{\bar{y}(k)\}$, $k \geq 0$, $k \in \mathbb{Z}$, which is a sequence of values $\{\bar{y}(0), \bar{y}(1), \cdots\}$ determined by the relation

$$\bar{y}(k) = y(t_k). \tag{1}$$

This is the ideal A/D (analog to digital) converter that samples $y(t)$ at times $t_0, t_1, t_2 \cdots$ producing the sequence $\{y(t_0), y(t_1), \cdots\}$ also denoted as $\{y(t_k)\}$.

D/A Converter: The D/A (digital to analog) converter receives as its input a sequence $\{\bar{u}(k)\}$, $k = 0, 1, 2, \cdots$ and outputs a (piecewise) continuous-time signal $u(t)$ determined by

$$u(t) = \bar{u}(k), \ \ t_k \leq t < t_{k+1}, \ k = 0, 1, 2, \cdots. \tag{2}$$

That is, this D/A converter keeps the value of $u(t)$ constant at the last value of the sequence entered, until a new value comes in. Such a device is called a zero-order hold (ZOH) device.

Higher-Order Hold

The ZOH device described above implements a particular procedure of *data reconstruction or extrapolation*. The general problem is as follows:

Given a sequence of real numbers $\{\bar{f}(k)\}$, $k = k_0, k_0 + 1, \cdots$ derive $f(t)$, $t \geq t_0$ so that

$$f(t_k) = \bar{f}(k), \ \ k = k_0, k_0 + 1, \cdots$$

Clearly, there is a lot of flexibility in assigning values to $f(t)$ in between the samples $\bar{f}(k)$; in other words there is a lot of flexibility in assigning the *intersample behavior* in $f(t)$.

A way to approach the problem is to start by writing a power series expansion of $f(t)$ for t, $t_k \leq t < t_{k+1}$, namely,

$$f(t) = f(t_k) + f^{(1)}(t_k)(t - t_k) + \frac{f^{(2)}(t_k)}{2!}$$
$$(t - t_k)^2 + \cdots$$

where $f^{(n)}(t_k) = \frac{d^{(n)} f(t)}{dt^n}|_{t=t_k}$, that is, the nth order derivative of $f(t)$ evaluated at $t = t_k$ (assuming that the derivatives exist).

Now if the function $f(t)$ is approximated in the interval $t_k \leq t < t_{k+1}$ by the constant value $f(t_k)$ taken to be equal to $\bar{f}(k)$, then

$$f(t) = f(t_k) \quad (= \bar{f}(k)), \ t_k \leq t < t_{k+1}$$

which is exactly the relation implemented by a ZOH. Note that here the zero-order derivative of the power series is used which leads to an approximation by a constant which is a zero-degree polynomial.

It is clear that more than the first term in the power series can be taken to approximate $f(t)$. If, for example, the first two terms are taken, then

$$f(t) = f(t_k) + f^{(1)}(t_k)(t - t_k)$$
$$= f(t_k) + \frac{f(t_k) - f(t_{k-1})}{t_k - t_{k-1}}(t - t_k)$$
$$= \bar{f}(k) + \frac{\bar{f}(k) - \bar{f}(k-1)}{t_k - t_{k-1}}(t - t_k)$$

for $t_k \leq t < t_{k+1}$, where an approximation for the derivative $f^{(1)}(t)$ has been used. The approximation between t_k and t_{k+1} is a ramp with slope determined by $f(t_k) = \bar{f}(k)$ and the previous value $f(t_{k-1}) = \bar{f}(k-1)$. Here the first-order derivative of the power series is used which leads to an approximation by a first-degree polynomial. A device that implements such approximation is called a *first-order hold (FOH)*. Similarly, we can define a *second-order hold*. Note that the formula of the above FOH is derived if we decide to use a first-degree polynomial to approximate $f(t)$ on $t_k \leq t < t_{k+1}$ and then enforce $f(t_k) = \bar{f}(k)$ and $f(t_{k-1}) = \bar{f}(k-1)$. This approach is known as *polynomial interpolation*.

Obtaining a continuous (or piecewise continuous) function from given discrete values may be seen as a *continualization procedure*. Contrast

this with the *discretization procedure* introduced by sampling earlier in this section.
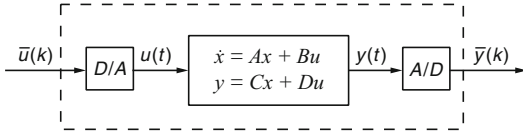
The continuous-time system with input $u(t)$ and output $y(t)$ together with the interface A/D and D/A converters can be seen as a system that receives a sequence of values $\{\bar{u}(k)\}$ as its input and produces a sequence of output values $\{\bar{y}(k)\}$. A digital controller can receive the system output $\{\bar{y}(k)\}$ as input and produce a $\{\bar{u}(k)\}$.

Quantization: The sampled output $\bar{y}(k) \in \mathbb{R}$ and it can take on an infinite number of values. In a digital device, however, a variable can take on only a finite number of values – this is because of the finite wordlength that is of the finite number of bits in the registers. So for $\{\bar{y}(k)\}$ to be used by a digital controller, an additional step is needed, that is, $\bar{y}(k)$ needs to be quantized. Under quantization, for example, values 2.315, 2.308, 2.3 with a 0.1 quantization step are all represented as 2.3. Quantization is an approximation and for short wordlengths, fewer number of levels, may lead to significant errors. Here we do not consider quantization.

Discrete-Time Models

Let a linear, continuous-time, time-invariant system be described by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{3}$$

If we consider some initial time t_k, its state response for $t \geq t_k$ is

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{4}$$

In view of (2), in a ZOH the input $u(t)$ will remain constant and equal to $u(t_k) (= \bar{u}(k))$ for a time period $t_{k+1} - t_k$. So

$$x(t) = e^{A(t-t_k)}\bar{x}(k) + \left[\int_{t_k}^{t} e^{A(t-\tau)}Bd\tau \right]\bar{u}(k), \tag{5}$$

where $\bar{x}(k) = x(t_k)$, $\bar{u}(k) = u(t_k)$. For $t = t_{k+1}$, (5) becomes

$$\bar{x}(k+1) = \bar{A}(k)\bar{x}(k) + \bar{B}(k)\bar{u}(k) \tag{6}$$

where $\bar{A}(k) \triangleq e^{A(t_{k+1}-t_k)}$ and $\bar{B}(k) \triangleq \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\tau)}Bd\tau$.

Consider now the output $y(t)$ and assume that it is sampled at times t_k' that do not necessarily coincide with the instants t_k at which the input is adjusted ($t_k \leq t_k' < t_{k+1}$). Then if $\bar{y}(k) \triangleq y(t_k')$,

$$\bar{y}(k) = \bar{C}(k)\bar{x}(k) + \bar{D}(k)\bar{u}(k), \tag{7}$$

where

$$\bar{C}(k) = Ce^{A(t_k'-t_k)}$$
$$\bar{D}(k) = C\left[\int_{t_k}^{t_k'} e^{A(t_k'-\tau)}d\tau \right]B + D.$$

In the case when all $k = 0, 1, 2, \cdots$, $t_k' = t_k$ and $t_{k+1} - t_k = T$ a constant period, called the *sampling period*. Then the sampled-data system is given by

$$\begin{aligned} \bar{x}(k+1) &= \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k) \\ \bar{y}(k) &= \bar{C}\bar{x}(k) + \bar{D}\bar{u}(k) \end{aligned} \tag{8}$$

where

$$\bar{A} = e^{AT}, \quad \bar{B} = \left[\int_0^T e^{A\tau}d\tau \right]B,$$
$$\bar{C} = C, \quad \bar{D} = D.$$

The intersample behavior of the continuous system can be determined using (5).

Example 1 Let the continuous-time system be given by (3) where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = [1 \quad 0], D = 0,$$

and let T denote the sampling period. The transfer function of the continuous-time system is $\hat{H}(s) = C(sI - A)^{-1}B = 1/s^2$, the double integrator. The discrete-time state-space representation of the system, which represents the continuous-time system preceded by a zero-order hold (D/A converter) and followed by a sampler [an (ideal) A/D converter], both sampling synchronously at a rate of 1/T, is given by $\bar{x}(k+1) = \bar{A}\bar{x}(k) + \bar{B}\bar{u}(k)$, $\bar{y}(k) = \bar{C}x(k)$, where

$$\bar{A} = e^{AT} = \sum_{j=1}^{\infty}(T^j/j!)A^j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}T = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix},$$

$$\bar{B} = \left(\int_0^T e^{A\tau}d\tau\right)B$$

$$= \left(\int_0^T \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}d\tau\right)\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} T & T^2/2 \\ 0 & T \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix},$$

$$\bar{C} = C = [1 \quad 0].$$

The transfer function (relating $\bar{y}$ to $\bar{u}$) is given by

$$\hat{H}(z) = \bar{C}(zI - \bar{A})^{-1}\bar{B}$$

$$= [1\ 0]\begin{bmatrix} z-1 & -T \\ 0 & z-1 \end{bmatrix}^{-1}\begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

$$= [1\ 0]\begin{bmatrix} 1/(z-1) & T/(z-1)^2 \\ 0 & 1/(z-1) \end{bmatrix}$$

$$\begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

$$= \frac{T^2}{2}\frac{(z+1)}{(z-1)^2}.$$

If we focus on single-input, single-output systems and consider ideal sampler A/D and ZOH D/A, then given the transfer function $G(s)$ of the continuous system, there is a direct formula to determine the transfer function of its discrete approximation $H(z)$, namely,

$$H(z) = (1 - z^{-1})Z\{G(s)/s\}. \qquad (9)$$

Here $Z\{G(s)/s\}$ means that first the inverse Laplace transform of $G(s)/s$ is taken to obtain $f(t) \triangleq [\mathcal{L}^{-1}(G(s)/s)]$. The function $f(t)$ is then sampled to obtain $f(kT), k = 0, 1, 2, \cdots$ and the z-transform of $f(kT)$ is evaluated. To illustrate, in the above example $G(s) = \frac{1}{s^2}$, $G(s)/s = \frac{1}{s^3}$, and $f(t) = \mathcal{L}^{-1}(\frac{1}{s^3}) = \frac{1}{2}t^2$, $t \geq 0$. Then

$$H(z) = (1 - z^{-1})Z\{\frac{1}{2}(kT)^2\}$$

$$= (1 - z^{-1})\frac{T^2}{2}Z\{k^2\}$$

$$= \frac{T^2}{2}\frac{z+1}{(z-1)^3}$$

as before.

Summary

Sampled-data systems arise in the digital control of systems and include both continuous and discrete-time dynamics. Discrete-time approximations of continuous-time systems using ideal samplers and ZOH devices were derived using state variable descriptions. Extensions include quantization and lead to hybrid dynamical systems which include both continuous and discrete variable dynamics.

A variation of the approach described in this entry of deriving sampled-data systems uses the discrete-time delta operator. This approach has the advantage that as the sampling period $T \rightarrow 0$, the discrete-time model reverts to the original continuous-time model, which is not the case with the more common approach described above.

Cross-References

▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions
▶ Linear Systems: Discrete-Time, Time-Invariant State Variable Descriptions

Recommended Reading

State variable and transfer function descriptions are covered in a variety of textbooks including Antsaklis and Michel (2006), Kailath (1980), Chen (1984), and DeCarlo (1989). For additional material on sampled-data systems, refer to Aström and Wittenmark (1990), Franklin et al. (1998), Jury (1958), and Ragazzini and Franklin (1958).

Bibliography

Antsaklis PJ, Michel AN (2006) Linear systems. Birkhauser, Boston

Aström KJ, Wittenmark B (1990) Computer-controlled systems: theory and design. Prentice-Hall, Englewood Cliffs

Chen CT (1984) Linear system theory and design. Holt, Rinehart and Winston, New York

DeCarlo RA (1989) Linear systems. Prentice-Hall, Englewood Cliffs

Franklin GF, Powell DJ, Workma ML (1998) Digital control of dynamic systems, 3rd edn. Addison-Wesley Longman Inc., Menlo Park, CA

Jury EI (1958) Sampled-data control systems. Wiley, New York

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Ragazzini JR, Franklin GF (1958) Sampled-data control systems. McGraw-Hill, New York

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice-Hall, Englewood Cliffs

Satellite Control

Finn Ankersen
European Space Agency, Noordwijk,
The Netherlands

Abstract

Spacecraft control systems are described for single and distributed space systems. The attitude dynamics is formulated including flexible and sloshing phenomena, followed by a description of attitude sensors and actuators. $\mathcal{H}_\infty$ and robust controls are formulated as signal-based two degree-of-freedom control architectures. The equations are given for the relative motion dynamics between spacecraft on elliptical orbits with the generic Yamanaka-Ankersen state transition matrix. Formulations are provided for rendezvous and docking scenarios and formation flying control, maneuvers, avionics, and laser metrology systems together with the onboard autonomy needs.

Keywords

Flexible modes; Formation flying; Fractionated spacecraft; $\mathcal{H}_\infty$ control; Multivariable systems; Relative dynamics; Rendezvous and docking; Robust control; Sloshing; Spacecraft attitude control; Spacecraft position control

Introduction

This entry explains the control needs of spacecraft after they have been separated from the launch vehicle and injected onto their initial orbit.

Actuators and sensors are explained followed by the control objectives. The state-of-the-art control techniques and architectures are addressed.

Spacecraft are classically well-known physical systems that can be described by first principles. The advantage is fairly precise plant models and uncertainty characterization of physical parameters. This is well suited for a model-based control design approach.

Mission Types

From a control point of view, space missions can be split into two main categories according to which physical states need to be controlled:

Attitude Control: This is needed by any spacecraft irrespective of the mission objectives. Such missions are typically low earth orbit (LEO) missions for astronomy, observations,

and, in higher orbits, constellations for navigation and communication. Further, there are interplanetary and planetary exploration science missions. The pointing requirements vary from a few degrees to milli-arc seconds.

Relative Position Control: Within distributed space systems, this is relevant for rendezvous and docking (RVD) and formation flying (FF) missions. It leads to a 6 degree-of-freedom (DOF) control problem as the relative attitude is also needed. The former is mostly for missions to space station logistics infrastructures and the latter for scientific missions. Relative position can also be required during the final stages of controlled planetary landings. Another category is missions with ultrahigh control performance requirements, where the spacecraft platform and the science instrument need to be considered as one coupled system.

Attitude Control

Fundamentally the three attitude angles $\boldsymbol{\theta}$ and angular rates $\boldsymbol{\omega}$ need to be controlled to a certain reference. See Fig. 1 for definition.

The general rigid body dynamics expressed in a rotating frame(∗), which is mostly the case when orbiting a central body, can be expressed as

$$\mathbf{N} = \frac{d^*(\mathbf{I}\boldsymbol{\omega}^*)}{dt} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}^* \qquad (1)$$

where $\mathbf{I}$ is the constant inertia matrix, $\boldsymbol{\omega}$ is the inertial angular velocity, and $\mathbf{N}$ is the torque acting on the spacecraft (Wie 1998).

The kinematics can be described by one of the 12 sets of Euler angles (can have singularities) or the hypercomplex quaternion vector (no singularities) (Hughes 1986).

The dynamics and kinematics equations need to be linearized and are in the general form of a coupled 12th order system. It is the fundamental model for the rigid body spacecraft control design.

Most modern spacecraft have large flexible appendices in the form of solar panels and large antennae reflectors. Fuel sloshing is a similar lightly damped oscillatory phenomena, which often needs to be taken into consideration. The incorporation of dynamic elements such as flexible panels, antennae, and sloshing fuel can be modeled by Eqs. (2) and (3) provided the overall rotation rate $\boldsymbol{\omega}$ and linear accelerations $\ddot{\mathbf{x}}$ are not too large.

$$\mathbf{M}_T \begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{N} \end{bmatrix} - \mathbf{L}\ddot{\boldsymbol{\eta}} \qquad (2)$$

$$\ddot{\eta}_k + 2\zeta_k \Omega_k \dot{\eta}_k + \Omega_k^2 \eta_k = -\frac{1}{m_k}\mathbf{L}^{\mathrm{T}} \begin{bmatrix} \ddot{\mathbf{x}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} \qquad (3)$$

Satellite Control, Fig. 1
Spacecraft body (*black*) and reference (*red*) frames. The frames coincide for $\boldsymbol{\theta} = \mathbf{0}$
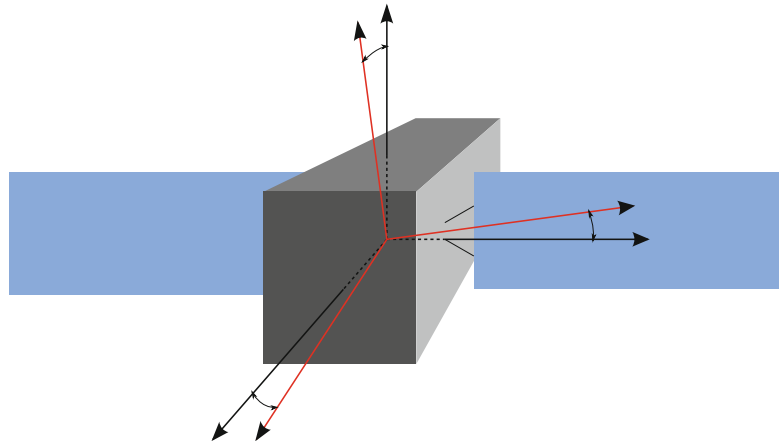
where

$\mathbf{M}_T$	: rigid body mass/inertia matrix
$\ddot{\mathbf{x}}, \dot{\boldsymbol{\omega}}$	: linear and angular acceleration
$\mathbf{F}, \mathbf{N}$	: forces and torques on the spacecraft
η_k	: the kth flexible state
ζ_k	: the kth flexible damping factor
Ω_k	: the kth flexible eigen frequency
m_k	: the kth modal mass (normalized to 1)
$\mathbf{L}$	: participation matrix of the kth mode

For attitude only the second row of Eq. (2) is needed, but translation is included here for the sake of completeness and later use.

The sensors utilized are typically gyroscopes for measuring the inertial angular rate, sun sensors to measure orientation at low accuracy, and star trackers for high-precision angular attitude measurements. All of those sensors are linear in their normal operational range and it suffices to use bias noise models for synthesis. Gyros do need a drift estimation and compensation to function properly over longer time. All sensors utilize redundancy for providing measurements around all three axes as well as providing fault tolerance. Some scientific observatory spacecraft use their telescopes for attitude measurements in order to obtain the required precision beyond the capability of star trackers.

The actuators producing pure torques are magnetic torquers, reaction wheels, and control momentum gyros. The last can produce large torques used for rapid slew maneuvers with little power. The last two types have nonlinear issues around low to zero speed due to friction issues. They accumulate angular momentum from asymmetric disturbances. This leads to a need for thrusters for angular momentum off-loading. Thrusters are also used to control the attitude directly on many spacecraft. They are mostly of on-off type, though continuous ones exist, and will need to be pulse width modulated (PWM) to obtain quasi-linear behavior. The nonlinear on-off nature needs to be taken into account for the control closed loop analysis. It is done by use of the negative inverse describing function (Ogata 1970) for stability analysis and nonlinear modeling for verification simulations in the time domain. For larger numbers of thrusters, an optimization-based selection algorithm is applied to the controller output.

Before using the plant model in Eq. (2) for a flexible spacecraft, a simpler multivariable model of a rigid spacecraft is used as in Eq. (4):

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{0} & \mathbf{B}_k \\ \mathbf{0} & \mathbf{A}_d \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_d \end{bmatrix} \mathbf{N} \qquad (4)$$

where $\mathbf{x} = [\theta_x, \theta_y, \theta_z, \omega_x, \omega_y, \omega_z]^{\mathrm{T}}$, $\mathbf{B}_k$ is identity, $\mathbf{B}_d = \mathbf{I}^{-1}$, and $\mathbf{A}_d$ is the general Jacobian for the dynamics having a real right half-plane (RHP) pole. See Ankersen (2011). The model describes the angular deviation from some reference frame, whose orientation can be arbitrary. It uses the Euler (3, 2, 1) rotation in the kinematics.

The state of the art of attitude control is today mostly based on $\mathcal{H}_\infty$ type of robust controllers with synthesis performed in the frequency domain. Requirements are often specified in the time domain, but formal methods exist to transform them into frequency domain weighting functions (ESA Handbook 2011) enhancing both synthesis and analysis. System uncertainties can be formulated as structured linear fractional transformations (LFT) with a general control configuration as illustrated in Fig. 2.

Commonly the $\mathcal{H}_\infty$ controller $\mathbf{K}$ is designed, and the lower loop in Fig. 2 is closed via a lower LFT such that $\mathbf{N} = F_l(\mathbf{P}, \mathbf{K})$ and robust stability (RS) and robust performance (RP) analysis is performed on the $\mathbf{N}, \boldsymbol{\Delta}$ system (Skogestad and Postlethwaite 1996).
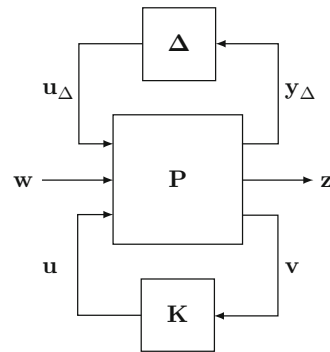
Satellite Control, Fig. 2 Robust control formulation, where $\boldsymbol{\Delta}$ is the structured uncertainty, $\mathbf{K}$ is the controller, $\mathbf{P}$ the partitioned formulation of the plant with weights, and $\mathbf{w}$ and $\mathbf{z}$ are exogenous inputs and outputs, respectively

On high performance pointing spacecraft, active vibration suppression of, e.g., cryocoolers is needed. The implementation of control design and recursive system identification can achieve significantly better attenuation compared to classical passive isolation techniques.

Lately optimization-based codesign of structures and control has been performed successfully. A joint performance function is formulated (mass, stiffness, pointing, fuel, etc.) and an optimization is performed (differential evolution algorithm) iterating on control design and finite element models (FEM). A μ-synthesis controller is synthesized, the pointing performance is fulfilled, and 15–20 % mass saving is obtained on the flexible structures. The entire process is fully automated (Falcoz et al. 2013).

Relative Position Control

For all distributed space systems, relative dynamics is important. Rendezvous and formation flying missions need tracking or maintenance of the desired relative separation, orientation, and position between or among the spacecraft. This is common and independent of the mission type and will be described in general terms ahead of the specific RVD and FF missions.

The general relative position dynamics between centers of mass (COMs) is in Eq. (5), where it is observed that the in-plane motion (x, z) is decoupled from the out-of-plane motion (y).

$$\ddot{x} - \omega^2 x - 2\omega\dot{z} - \dot{\omega}z + k\omega^{\frac{3}{2}}x = \frac{1}{m_c}F_x$$
$$\ddot{y} + k\omega^{\frac{3}{2}}y = \frac{1}{m_c}F_y \quad (5)$$
$$\ddot{z} - \omega^2 z + 2\omega\dot{x} + \dot{\omega}x - 2k\omega^{\frac{3}{2}}z = \frac{1}{m_c}F_z$$

where $\omega = \omega(t)$ is the orbital angular rate, m_c is the chaser mass, F_{xyz} is the force on the chaser, and k is a constant determined by the orbit and is valid for any Keplerian orbit with eccentricity $\varepsilon < 1$.

The Yamanaka-Ankersen equations (Yamanaka and Ankersen 2002) provide the generalized homogeneous solution in the form of the transition matrix $\boldsymbol{\Phi}$, where the solution can be written as

$$\mathbf{x}(t) = \boldsymbol{\Lambda}^{-1}(\nu)\boldsymbol{\Phi}(\nu)\boldsymbol{\Phi}_0^{-1}(\nu_0)\boldsymbol{\Lambda}(\nu_0)\mathbf{x}(t_0) \quad (6)$$

where ν is the orbital true anomaly and Λ are transformation matrices to and from the time domain. The elements of $\boldsymbol{\Phi}$ in Eq. (6) are detailed in (Ankersen 2011), where relevant particular solutions are also to be found. Equation (6) reduces to the well-known Clohessy-Wiltshire equations for circular orbits ($\varepsilon = 0$) (Clohessy and Wiltshire 1960). Equation (6) is used for feedforward control and trajectory propagation in the guidance function. During the final approach (see Fig. 3), a model accounting for the docking port-to-port relative position and the couplings from the relative attitude to the position is utilized and formulated in Eqs. (7) and (8) (Ankersen 2011):

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_c \end{bmatrix}\mathbf{x} + \begin{bmatrix} \mathbf{B}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_c \end{bmatrix}\mathbf{u} \quad (7)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{B}_{dc_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{B}_{dc_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}\mathbf{x} \quad (8)$$

where $\mathbf{x} = [\mathbf{x}_p, \dot{\mathbf{x}}_p, \boldsymbol{\theta}_c, \boldsymbol{\omega}_c]^{\mathrm{T}}$, $\mathbf{y} = [\mathbf{x}_{pp}, \dot{\mathbf{x}}_{pp}, \boldsymbol{\theta}_c, \boldsymbol{\omega}_c]^{\mathrm{T}}$, index p refers to COM positions, index c to chaser attitude, index pp to port-to-port position, and $\mathbf{B}_{dc_1}, \mathbf{B}_{dc_2}$ are the coupling matrices of the docking port.

A relative motion scenario for a typical RVD mission looks like in Fig. 4. During the final approach (<300 m range), the chaser relative attitude and relative position are controlled. During the other phases, the chaser attitude is Earth pointing and the relative position is controlled at the station-keeping (SK) points, $s_0, \cdots, s_4$ in Fig. 4. The trajectories are typically open loop feedforward controlled (often with midcourse corrections).

The avionics sensors for the attitude control part are generally similar to those described earlier under attitude control in connection with Fig. 1. Active laser CCD type of sensors

Satellite Control, Fig. 3
Definition of
COM-to-COM and
port-to-port positions, **s**
and $\mathbf{s}_{pp}$, respectively,
between two spacecraft
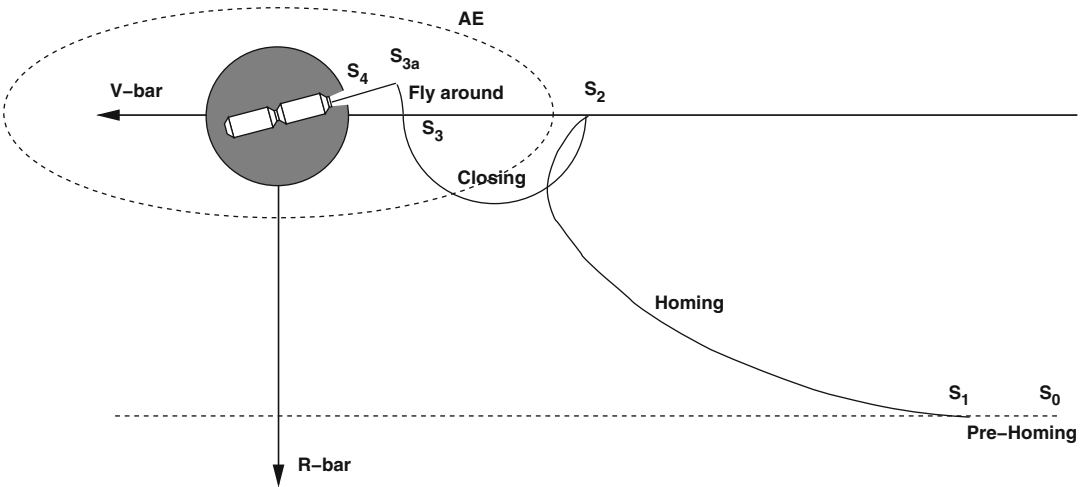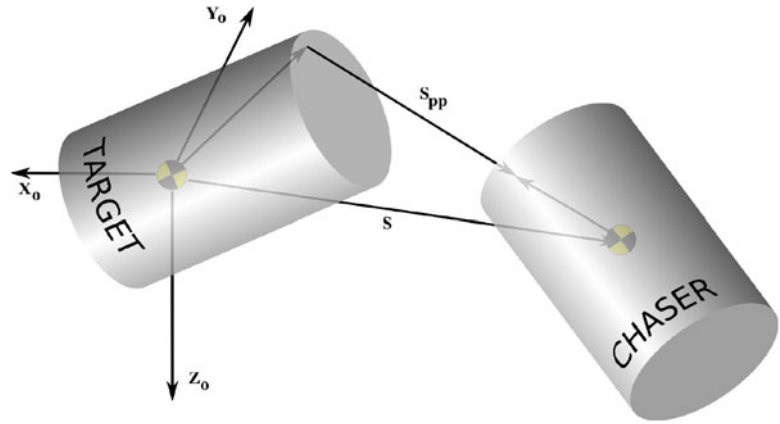
Satellite Control, Fig. 4 This figure shows the phases of typical relative motion approach. The *shaded area* is a keep-out zone (KOZ) defined for safety reasons. V-bar is the x-axis and R-bar is the z-axis

is used to measure the relative position (range and line-of-sight (LOS) angles) and at short range (<50 m) the relative attitude. They require a target pattern to provide precise measurements at short range. Accelerometers are used, particularly for pulsed maneuvers. The next generation of RVD GNC systems, test flown, will utilize Lidar, infrared cameras, and visual cameras in combination with advanced image processing providing RVD capabilities with both cooperative and passive target spacecraft.

The actuators are mostly thrusters arranged to achieve controllability for all the 6DOF maneuvers needed. Based upon the controller output, the active thrusters are selected by means of some

type of fuel optimization algorithm. The selected thrusters are then pulse width modulated (PWM) within the sampling time.

The controllers are frequently of multivariable $\mathcal{H}_\infty$ type. They are similar to what is described in connection with Fig. 2. Flexible modes and in particular sloshing need to be taken into account using Eq. (2). Sloshing pendulum models are used during boost maneuvers and spring mass damper models during other modes. The couplings between relative attitude and relative position in Eq. (8) can be analytically decoupled setting the matrix $\mathbf{C}$ to identity and premultiplying with a decoupling matrix $\mathbf{V}_d$, such that

$$\mathbf{V}_d \mathbf{C} = \mathbf{I} \Leftrightarrow \mathbf{V}_d = \mathbf{C}^{-1} \qquad (9)$$

Satellite Control, Fig. 5
Principal structure of the 2
degree-of-freedom
controller
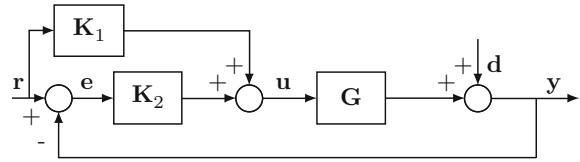
and by the inversion theorem for partitioned matrices the upper right partition just changes sign. The designed controller then needs to be premultiplied by $\mathbf{V}_d^{-1}$, which facilitates a simpler control design maintaining the 6DOF performance after 2 times 3DOF synthesis.

A 2 degree-of-freedom control architecture as in Fig. 5 is beneficial since much of the performance is achieved by controller $\mathbf{K}_1$. The structure of the synthesis formulation is a signal-based model-reference configuration for the $\mathcal{H}_\infty$ control rather than the more classical mixed sensitivity type. It has proven to have higher robustness and performance for this type of applications. As an example, consider a controller that has to follow a sawtooth motion of the docking port of the International Space Station (ISS) with an amplitude of 0.4 m and reversal times of 8 s. The signal-based model-reference controller manages to track such a motion with errors less than 0.01 m compared to the best operational performance of 0.08 m.

Formation flying usually includes more than two spacecraft with the need to be controlled relative to each other. The objective of FF is to form an instrument in space, not possible with fixed structures, like a synthetic aperture or an interferometer of large size.

The performance needs are high and require innovative high-precision ($<1\,\mu$m) metrology sensors. They are based on divergent laser beams for the coarse part to be able to transit from lower to higher accuracy. The fine metrology uses a laser beam and internal interferometers to reach the μm domain. Actuators are in the range of μN thrust, which can be achieved with either cold gas or electrical propulsion thrusters.

The maneuvers realized by entire formations are rotation, resizing, and slew while maintaining the formation in most cases (Alfriend et al. 2010).

Formation flying missions with the highest performance requirements have optical payloads, which need to have internal control loops at component level. To reach the performance required for applications such as optical interferometry, the formation and payload must be considered as one system. The synthesis of a multivariable controller then handles all the cross couplings in the system needed to reach performance. Beyond flexible modes, such systems might also have a need for active vibration damping for systems using cryocoolers.

The GNC architecture is often centralized for nominal science operational modes. For the formation deployment and contingency situations, a decentralized control architecture is needed. This leads to a dual architecture GNC system in general for formation flying systems. The onboard autonomy needs to be fairly high in order to cope with the contingencies in the formation without ground intervention.

Finally there is an emerging concept of fractionated spacecraft. There, a formation consists of a large number of small simple vehicles maneuvering relative to each other fully autonomously based upon the nearest neighbor knowledge and not necessarily information about the entire formation (Cornford 2012).

Summary and Future Directions

The control of spacecraft has been described for pure attitude control needs and for spacecraft performing relative proximity maneuvers like rendezvous and formation flying. The focus has been on sensors, actuators, dynamics, and the robust control methods applied today.

The further development direction of the field is expected to be increased on board autonomy with replanning capabilities and fault-tolerant

GNC designs. Model predictive control (MPC) will enter in particular on the guidance functions. More integrated GNC system-level designs, of multidisciplinary nature, are expected.

Cross-References

▶ Fault-Tolerant Control
▶ H-Infinity Control
▶ Model-Predictive Control in Practice
▶ Nominal Model-Predictive Control

Bibliography

Alfriend K, Vadali S, Gurfil P, How J, Breger L (2010) Spacecraft formation flying. Elsevier, Amsterdam/Boston/London

Ankersen F (2011) Guidance, navigation, control and relative dynamics for spacecraft proximity maneuvers. Aalborg University, Denmark. ISBN:978-87-92328-72-4

Bryson A (1999) Control of spacecraft and aircraft. Princeton University Press, Princeton

Clohessy W, Wiltshire R (1960) Terminal guidance system for satellite rendezvous. J Aerosp Sci 27(9): 653–658

Cornford S (2012) Evaluating a fractionated spacecraft system: a business case tool for DARPA's F6 program. In: Aerospace conference, Big Sky. IEEE, Big Sky, MT, pp 1–20

D'Errico M (2012) Distributed space missions for earth system monitoring. Springer, New York

ESA Handbook (2011) ESA pointing error engineering handbook. European Space Agency. http://peet.estec.esa.int

Falcoz A, Watt M, Yu M, Kron A, Menon P, Bates D, Ankersen F, Massotti L (2013) Integrated control and structure design framework for spacecraft applied to BIOMASS satellite. In: 19th IFAC conference on automatic control in aerospace, Würzburg, Germany, 2–6 Sept 2013

Fehse W (2003) Automated rendezvous and docking of spacecraft. Cambridge University Press, Cambridge/New York

Hughes P (1986) Spacecraft attitude dynamics. Wiley, New York

Kaplan M (1976) Modern spacecraft dynamics & control. Wiley, New York

Ogata K (1970) Modern control engineering. Prentice-Hall, Englewood Cliffs

Sidi M (2000) Spacecraft dynamics and control: a practical engineering approach. Cambridge University Press, Cambridge

Skogestad S, Postlethwaite I (1996) Multivariable feedback control. Wiley, Chichester/New York

Wertz J (1980) Spacecraft attitude determination and control. Kluwer, Dordrecht

Wie B (1998) Space vehicle dynamics and control. American Institute of Aeronautics and Astronautics, Reston

Yamanaka K, Ankersen F (2002) New state transfer matrix for relative motion on an arbitrary elliptical orbit. J Guid Control Dyn 25(1):60–66

Scheduling of Batch Plants

John M. Wassick
The Dow Chemical Company, Midland, MI, USA

Abstract

For manufacturers operating batch plants, production scheduling is a critical and challenging problem. A thorough understanding of the problem and the variety of solutions approaches is needed to achieve a successful application. This entry will present a brief overview of batch operations and the state of the art of batch plant scheduling for nonexperts in the field.

Keywords

Dispatching rules; Optimization; Process networks; Production sequencing; Product wheel

S

Introduction

Batch plants, manufacturing operations composed of unit operations that operate in batch mode, are the primary manufacturing operations for the production of high margin products such as pharmaceuticals, specialty chemicals, and advanced materials. The scheduling of the sequence of operations over time has a significant impact on the overall performance of a batch plant (White 1989). The economic importance of batch plants, and the importance of scheduling for batch plants, has spawned a large body of

research on the topic and a variety of commercial offerings.

The Nature of Batch Plants

In batch operations, the material transformation takes place in stages and the operation of each stage occurs over a specified time while the material remains in a particular unit operation performing that stage of production. (A familiar batch operation is baking a cake. Ingredients and their amounts, specified by a recipe, are combined and then subjected to a constant temperature over specified period of time to produce a cake.) A batch plant may have parallel units for some stages. Other stages may be operated in a continuous flow mode with a storage unit feeding the stage and another storage unit receiving the stage output. The path through the unit operations may be product dependent. Batch plants have highly diverse operational characteristics.

There are two broad categories of batch processes: (1) sequential where a batch moves from one stage to another without losing its identity and (2) networked where batches can be combined or split to feed downstream units (Mendez et al. 2006). Sequential processes can be further classified as single stage, multi-stage, or multipurpose.

The nature of a batch process and the different process structures can be explored by referring to the process depicted in Fig. 1 (Chu et al. 2013). As drawn, this batch plant operates as a multi-stage sequential process where a batch starts in raw material preparation stage (selected raw materials are loaded and then blended for a specified time), moves to the reaction stage with two parallel units (prepared raw materials plus additives react at a constant temperature for a specified period of time), moves to the finishing stage (intermediate product is subjected to a vacuum for a specified period of time to remove volatile by-products), and finally is processed in the drumming stage (finished product is packaged in drums). If finished product storage tanks were placed between finishing and drumming to allow the drumming stage to be scheduled independently of the first three stages, then the drumming operation would represent a single stage sequential process. If we further assume that for some finished products Reactor 1 produces a batch of precursor for Reactor 2 and that some products produced in the reactors bypass the finishing stage and go directly to drumming, then the underlying plant would be a multi-purpose sequential process. Finally, if intermediate storage tanks exist for storing multiple batches of the precursors produced by Reactor 1 and the contents of the tanks are drawn off to produce multiple, subsequent batches in both reactors then the underlying plant is a networked process.

Besides the general structure of a batch plant, the specific processing requirements, resources needs, and process constraints have significant impact on the complexity of the scheduling problem. One important aspect is limited resources that are shared between different operations. The availability and capacity of shared resources place a severe constraint on the timing of competing operations. Another significant factor is intermediate storage between
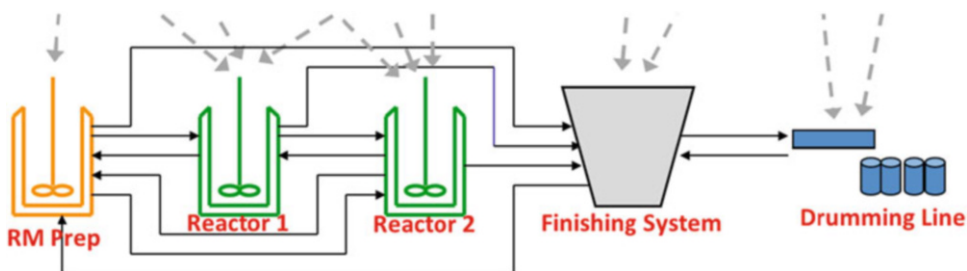
Scheduling of Batch Plants, Fig. 1 Example batch plant (*Solid lines* represent material flows from limited inventory. *Dashed lines* represent material flow from unlimited inventory)

stages and the inventory policies that are enforced. Like shared resources, intermediate storage places hard constraints on the timing of upstream and downstream stages, especially when no storage is available. A third important constraint on scheduling is product transition policies that dictate what operations need to be performed to move from one product to another in a given stage. Such operations, sometimes called setups, might involve cleaning, or producing buffer batches to isolate the chemistry of one product from another. These operations involve costs and subtract from the productive use of the equipment so they have significant impact on the sequencing of products through the plant.

Production Scheduling of Batch Plants

Production scheduling in a batch plant involves three fundamental decisions: (1) determining the size of each batch in each stage, (2) assigning a batch to a processing unit in each stage, and (3) determining the sequence and timing of processing on each unit. These decisions are well illustrated by a graphical planning board or Gantt chart as shown in Fig. 2 (Chu et al. 2013). Personnel charged with creating and managing production schedules often rely on such a graphical tool to construct, analyze and report the schedule. Generally production schedules are determined using the information listed in Table 1.

The scope of the scheduling decisions is defined by the level of process detail considered in the scheduling problem. This idea can be examined by referring to Figs. 1 and 2. Such a Gantt chart could apply to a batch plant with four stages of production: raw material preparation, reaction, finishing, and drumming, with two parallel reactors in the reaction stage. If dedicated finished product storage exists with large enough capacity to cover the process lead time then one schedule could be confined to the first three stages of production and a different schedule applied to the drumming stage. The scope of the scheduling problem could be further reduced if raw material
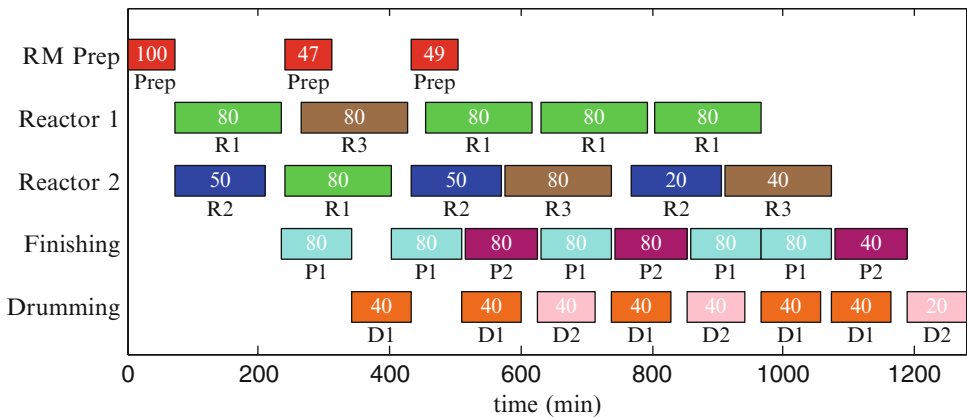
Scheduling of Batch Plants, Fig. 2 Gantt chart of a production schedule

Scheduling of Batch Plants, Table 1 Information generally used to construct a production schedule

Scheduling information	Examples
Detailed production recipes	Batch times, processing rates, unit ratios, sequence dependencies
Equipment data	Capacities, availabilities, product suitability
Facility information	Shared resource availability and capacities, storage capacities
Production costs	Raw materials, utilities, setups, cleanings, manpower
Production targets	Inventory replenishments, customer orders with due dates
Current process status	Current inventories, operations in progress, schedule items fixed in future time

preparation only takes place just in time to load a reactor rather than execute as soon as possible. In this situation, the time for raw material preparation could be added to the reactor batch time and the schedule would involve only the reactors and the finishing system with the raw material unit or units schedule implied by the reactor schedule. At a higher level still, the first three stages of production could be considered a production train and scheduling could then be reduced to planning campaigns of batches for each product over time with the detailed synchronization of the individual stages left to operations personnel. Obviously with each level of abstraction some efficiency in the schedule is lost and subsequently the opportunity to increase throughput of the plant.

In most batch plants a person with a title such as "production scheduler" is charged with the scheduling decisions. In general, the production scheduler is responsible for delivering a production schedule that meets customer orders on time and maintains finished product inventory while dealing with rush orders, late deliveries, equipment breakdowns and other contingencies. Generally schedulers develop and publish a schedule to manufacturing on a regular basis (e.g., every 2 days, once a week, etc.) and then monitor ongoing circumstances (e.g., actual production vs. plan, new demand, etc.) to determine if minor adjustments to the schedule are needed or if a complete new schedule needs to be published. The construction of a schedule can be an iterative process involving negotiations with manufacturing, supply chain, sales, maintenance and logistics. The tools available to the production scheduler can have a significant impact on the quality of schedules they produce.

It is evident from the description above that production scheduling of batch plants is really carried out as an exercise in rescheduling in response to disturbances identified through feedback from the process and market. Under these circumstances production scheduling serves as a form of high level feedback control of the process. In this regard the manipulated variables are the production amounts for each product and the controlled variables are the inventory levels and customer service levels for each product. A scheduling problem can be converted to a state-space formulation and compared to model predictive control (Subramanian et al. 2012).

Solution Approaches

The solution approaches applied to scheduling batch plants cover a wide spectrum of sophistication. A very simple form is nothing more than a sequence of batches maintained on a white board in the plant control room. A level above this would be the use of custom spreadsheets for arranging batches chronologically and computing finished product inventory. Another step up is the use of a manually manipulated Gantt chart as illustrated in Fig. 2, possibly pre-populated by an automated planning application that determines the volume to be produced across the units of production while leaving the detailed sequencing and timing decisions to the production scheduler. The highest level of sophistication involves an automatically generated schedule with the application retrieving all the necessary data from the appropriate business databases and plant control system.
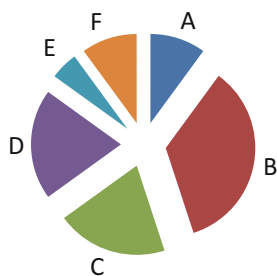
Regardless of the level of sophistication, all solution approaches rely on two fundamental components for developing a schedule. One is the modeling paradigm used to represent the physical system in a more abstract way. The primary components are: material balances in terms of batches or units of measure (e.g., pounds), and timing information as either precedence-based describing the order of operations or time grid-based describing the instant at which any operation takes place. Time can either be described by a continuous representation or divided into discrete increments. Within these two aspects of the modeling framework, significant freedom exists to describe the scheduling problem. The second fundamental component is the solution method used to generate the schedule. Each method has its strengths; therefore solutions combining methods are also used. The essential problem is to produce the information needed to draw the Gantt chart in Fig. 2 given the information in Table 1.

Product Wheel

While the primary objective of production scheduling is to meet customer orders while managing finished product inventory, other operational issues need to be managed, such as minimizing product transition costs, minimizing variability in manufacturing operations, keeping the scheduling process simple, and balancing the tradeoff between production lead times, inventory, and transition losses. The product wheel is a practical approach widely used in industry to address these competing issues. A product wheel is a regular repeating sequence of products made on a specific unit operation or an entire production process. A product wheel is typically depicted as a pie chart as shown in Fig. 3. Segments of the pie, called spokes of the wheel, represent a production campaign of a particular product. The size of the spoke represents the length of the campaign relative to the overall duration, or cycle time, of the wheel.

A product wheel has specific design parameters to address various operations objectives. The sequences is fixed and optimized for minimum transition costs. The overall cycle time is fixed and optimized to balance lead time and inventory costs. The campaign size or spokes for each product are sized to match average demand for each product. The fixed pattern of the product wheel provides manufacturing with a predictable operational rhythm and the production scheduler with a very structured decision framework. Refer to King and King (2013) for a complete treatment of product wheels.

In practice, the duration of a campaign for a given product will vary from cycle to cycle as it will be sized to replenish any inventory consumed in the previous cycle. Low volume products may not be made on every cycle, although they will have a fixed location in the sequence. This same approach applies to make-to-order products that are not inventoried but produced to fill specific orders. Thus, in some cases a product wheel may be composed of several different but repeating cycles.

Dispatching Rules Used in Discrete Manufacturing

Batch processes are closely related to discrete manufacturing. Batches processed on a unit are analogous to jobs processed on a machine. Much of the literature on machine scheduling has focused on the analysis of the specifics encountered in general classes of problems such as single machines, parallel machines, flow shops and job shops, and developing constructive scheduling rules where a schedule is built up by adding one job at a time (Blackstone et al. 1982). Under certain circumstances these rules used for machine scheduling can be applied to scheduling batch plants. This allows one to take advantage of a great body of literature, and at times, very simple scheduling rules that have proven optimality or worst case performance limits.

Consider again the batch process referred to in Fig. 1 which has two parallel reactors. The two reactors can be modeled as a single stage process and scheduled like parallel machines using the simple *shortest processing time first* (SPT) rule if the following circumstances hold: (1) raw material preparation can be included in the batch time of reactors, (2) significant storage exists between the reactors and finishing to essentially isolate the two stages, (3) product specific batch times are identical for both reactors, (4) the number of batches of each product is given (perhaps the result of an inventory policy for make to stock products), and (5) the objective is to minimize the total completion time for all batches. The SPT rule is simply to select, whenever a reactor is free, the batch with the shortest processing time from those yet to be processed. This can be proven to produce an optimal schedule for the given conditions.

Scheduling of Batch Plants, Fig. 3 Product wheel

Another simple dispatching rule to mention is the *earliest due date first* (EDD) rule. This rule is designed for single stage processes without parallel units where each batch has an associated due date. The rule simply orders the batches in increasing order of their due dates to minimize the maximum lateness of all orders.

The conditions needed for the SPT rule or the EDD rule to produce an optimal schedule can be quite restrictive when considering batch processes, however these rules and others found in the machine scheduling literature (Baker and Trietsch 2009) can still produce a good initial schedule even in cases where optimality conditions are not satisfied. Once generated, the schedule can be improved by manual manipulation of the Gantt chart or the application of improvement heuristics.

Improvement Heuristics

Improvement heuristics try to improve the current schedule by searching for alternative solutions either in the neighborhood of the current schedule or by broadly exploring the solution space. The behavior of these algorithms is determined by tuning parameters that balance the use of the two search techniques and the underlying algorithm that performs the search. Improvement heuristics generally have the following basic procedure:

Step 1: Initialize – determine a starting schedule

Step 2: Generate alternatives – build modifications to the current schedule

Step 3: Check for improvements in modified schedule – if no improvement is found return to Step 2 otherwise proceed to Step 4

Step 4: Check for termination – terminate the algorithm if the number of iterations is exceeded or minimal improvement is obtained.

Many improvement heuristics are inspired by processes found in nature. Two of the more popular heuristics are simulated annealing which mimics the crystal formation during the cooling process of dense matter (Ryu et al. 2001) and genetic algorithms that mimic the evolution of a species over time (Löhl et al. 1988). A key aspect of improvement heuristics is the representation of the schedule in context of the algorithm used. For problems with complicated constraints this becomes a challenge. Nevertheless, when tuned

properly and used where they fit the problem, improvement heuristics can produce very good schedules quickly.

Tree Search Methods

The scheduling solutions considered so far have taken a relatively simple view of a batch process as a single stage process or a flow shop. In situations where a batch plant involves shared resources, complicated transition rules or is a process network, tree search methods are better suited because they can deal with a large number of degrees of freedom and many types of constraints. Tree search methods rely on representing alternative schedules as the final nodes in a tree where intermediate nodes represent partial solutions of the schedule. To be practical, these methods must be able to effectively search through the tree while pruning non promising branches (see Fig. 4). Three of the most popular techniques are mathematical programming, constraint programming, and beam search.

Mathematical programming solution techniques for scheduling generally convert the problem to a mixed integer linear programming (MILP) formulation where branching at nodes of the tree represent alternative values of the integer or binary variables. The tree is searched by a branch-and-bound algorithm which eliminates a node and the branch that emanates from it if the lower bound of the objective function represented by the terminal nodes of the branch is larger than the current best schedule. The MILP formulation can be stated generically as

$$\begin{aligned} \min \quad & z = cx + fy \\ \text{s.t.} \quad & Ax + By \geq b \\ & x \in \Re_+^n, y \in \{0, 1\}^p \end{aligned}$$

where c, f, b are vector of constants, A and B are matrices of constants, and the solution is defined
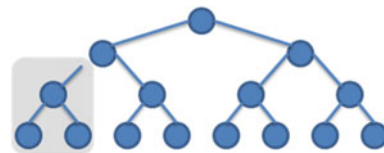
Scheduling of Batch Plants, Fig. 4 Trimming the solution tree

by the vector variables x and y. A key feature of using mathematical programming is to represent the relationships implied in Table 1 and Fig. 1 in terms of algebraic descriptions. The advantage of this approach is that a proven optimal solution exists for a problem stated this way. This provides the means to assess the quality of the solution and the impact of implementing the solution. The drawback of this approach is that since binary variables are used to represent the assignment of a batch to a processing unit, and the sequence and timing of processing on each unit, their number grows rapidly with the number of units and the length of the scheduling horizon. However, the performance of modern computing hardware and commercial solvers for MILP problems has allowed industrial size problems to be tackled.

A large variety of modeling paradigms have been developed to produce a MILP solution (Floudas and Lin 2004; Mendez et al. 2006). They address both sequential and networked processes using continuous time or discrete time representations. For sequential processes, time slot approaches have been developed. For networked processes, the resource task network and the state task network have been investigated by many researchers and have been used in industrial applications.

Constraint programming (CP) formulates a problem by writing constraints; but unlike the MILP method, the CP method stresses the feasibility of solutions rather than optimality. Another important difference is that constraints in the CP method do not have to be formulated as algebraic relationships but can be a more general form, thus making it easier in CP to represent complicated constraints. CP processes the constraints sequentially to reduce the space of possible solutions. At each node in the tree, CP processes one constraint after another, reducing the search space at each constraint. Being much newer than mathematical programming, constraint programming has a smaller body of literature to review but excellent performance has been reported in the literature (Baptiste et al. 2001).

In the beam search method, the branch-and-bound algorithm is modified to only evaluate the most promising nodes at any given level of the search tree (Ow and Morton 1988). The number of nodes evaluated is called the beam width and it is a key tuning parameter of the method. Another important element of the method is the technique used to retain nodes for complete evaluation. The technique must balance speed versus thorough evaluation to keep the method practical without discarding promising nodes. The beam search method applied to scheduling has been investigated by many authors (Sabuncuoglu and Bayiz 1999).

Simulation

The simulation approach to scheduling batch plants relies on representing the plant and the relationships inferred by Table 1 in a computer program whose algorithms recreate the behavior of the plant when executed. Generally, the simulators used for batch operations apply discrete event simulation (DES) where entities that have attributes like size, due date, priority, etc. are operated on by activities for a specified duration. Fundamental to DES are the use of queues to hold entities until conditions in the simulation allow them to proceed to their next activity. Time in a DES does not proceed in a continuous manner but rather advances when activities occur. Simulation has the advantage of being able to describe processes and operating policies of arbitrary complexity and model variability in the process operation. Simulators can be used to evaluate manually created schedules or can be combined with optimization and heuristics to produce schedules by simulation-based optimization (Pegden 2011).

An alternative to DES for batch scheduling is the use of multi-agent simulators which are composed of semiautonomous agents assigned to represent the operation of the process and the associated decision making. Each agent has a local goal and communicates with other agents to accomplish it. Like DES, multi-agent simulators are capable of describing very complicated processes. A production schedule can be built through negotiations between agents (Chu et al. 2013).

Selecting a Solution Approach

The selection of the approach for a given batch plant should be value-based, balancing improved revenue with long term cost of ownership by

considering such factors as the technical competency of the production scheduler, the expected capacity utilization of the plant, the operational complexity of the plant, and the cost to maintain the scheduling application. The key is to obtain the least complicated solution by reducing the scheduling problem to the highest level of abstraction and by using the simplest solution method that provides an effective schedule. See Harjunkoski et al. (2013) and Pinedo (2008) for a survey of methods and recommendations for their practical application.

Summary and Future Directions

While there are a great variety of solution methods for scheduling, there are still promising research areas to be investigated. The recent introduction of sophisticated, object oriented process control systems with ties to enterprise management systems sets the stage for the development of automatic, real time scheduling. It is here that the principles of feedback control can be applied to batch plant scheduling. Pursuit of this goal will require continued development of fast, adaptive scheduling methods, real time assessment techniques of schedule performance, and tight integration of scheduling with the process control.

Cross-References

▶ Control and Optimization of Batch Processes
▶ Models for Discrete Event Systems: An Overview

Bibliography

Baker KR, Trietsch D (2009) Principles of sequencing and scheduling. Wiley, Hoboken
Baptiste P, Le Pape C, Nuijten W (2001) Constrained-based scheduling: applying constraint programming to scheduling problems. Kluwer Academic, Dordrecht
Blackstone JH, Phillips DT, Hogg GL (1982) A state-of-the art survey of dispatching rules for manufacturing job shop operations. Int J Prod Res 20:27–45
Chu Y, Wassick JM, You F (2013) Efficient scheduling method of complex batch processes with general network structure via agent-based modeling. AIChE J. doi:10.1002/aic.14101 (accepted)
Floudas CA, Lin XX (2004) Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Comput Chem Eng 28:2109–2129
Harjunkoski I, Maravelias C, Bongers P, Castro P, Engell S, Grossmann I, Hooker J, Méndez C, Sand G, Wassick J (2013, submitted) Scope for industrial applications of production scheduling models and solution methods. Comput Chem Eng 60:277–296
King PL, King JS (2013) The product wheel handbook: creating balanced flow in high-mix process operations. Productivity Press, New York
Löhl T, Schulz C, Engell S (1988) Sequencing of batch operations for a highly coupled production process: genetic algorithms versus mathematical programming. Comput Chem Eng 22:S579–S585
Mendez CA, Cerda J, Grossmann IE, Harjunkoski I, Fahl M (2006) State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput Chem Eng 30:913–946
Ow PS, Morton TE (1988) Filtered beam search in scheduling. Int J Prod Res 26:35–62
Pegden DD (2011) Business benefits of Simio's risk-based planning and scheduling (RPS). In: Simio – resources – white papers. http://www.simio.com/resources/white-papers/. Accessed 1 June 2013
Pinedo ML (2008) Scheduling: theory, algorithms, and practice, 3rd edn. Springer, New York
Ryu JH, Lee HK, Lee IB (2001) Optimal scheduling for a multiproduct batch process with minimization of penalty on due date period. Ind Eng Chem Res 40:228–233
Sabuncuoglu I, Bayiz M (1999) Job shop scheduling with beam search. Eur J Oper Res 118:390–412
Subramanian K, Maravelias CT, Rawlings JB (2012) A state-space model for chemical production scheduling. Comput Chem Eng 47:97–110
White CH (1989) Productivity analysis of a large multiproduct batch processing facility. Comput Chem Eng 13:239–245
Wong TN, Leungy CW, Mak KL, Fung RYK (2006) Integrated process planning and scheduling/rescheduling – an agent-based approach. Int J Prod Res 44:3627–3655

Singular Trajectories in Optimal Control

Bernard Bonnard[1] and Monique Chyba[2]
[1]Institute of Mathematics, University of Burgundy, Dijon, France
[2]University of Hawaii-Manoa, Manoa, HI, USA

Abstract

Singular trajectories arise in optimal control as singularities of the end-point mapping. Their importance has long been recognized, at first in the

Lagrange problem in the calculus of variations where they are lifted into abnormal extremals. Singular trajectories are candidates as minimizers for the time-optimal control problem, and they are parameterized by the maximum principle via a pseudo-Hamiltonian function. Moreover, besides their importance in optimal control theory, these trajectories play an important role in the classification of systems for the action of the feedback group.

Keywords

Abnormal extremals; End-point mapping; Martinet flat case in sub-Riemannian geometry; Pseudo-Hamiltonian

Introduction

The concept of singular trajectories in optimal control corresponds to *abnormal extrema* in optimization. Suppose that a point $x^* \in X \simeq \mathbb{R}^n$ is a point of extremum for a smooth function $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}$ under the equality constraints $F(x) = 0$ where $F : X \to Y$ is a smooth mapping into $Y \simeq \mathbb{R}^p$, $p < n$. The *Lagrange multiplier rule* (Agrachev et al. 1997) asserts the existence of nonzero pairs (λ_0, λ^*) of Lagrange multipliers such that $\lambda_0 \mathcal{L}'(x^*) + \lambda^* F'(x^*) = 0$. The *normality condition* is given by $\lambda_0 \neq 0$, and the abnormal case corresponds to the situation when the rank of $F'(x^*)$ is strictly less than p.

Abnormal extremals have played an important role in the standard calculus of variations (Bliss 1946). Indeed, consider a classical Lagrange problem:

$$\frac{dx}{dt}(t) = F(x(t), u(t)), \ \min_{u(.)} \int_0^T L(x(t), u(t)) dt$$
$$x(0) = x_0, x(T) = x_1,$$

where $x(t) \in X \simeq \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, F and L are smooth. Using an infinite dimensional framework, the Lagrange multiplier rule still holds and an abnormal extremum corresponds to a singularity of the set of constraints.

Definition

Consider a system of $\mathbb{R}^n$: $\frac{dx}{dt}(t) = F(x(t), u(t))$ where F is a smooth mapping from $\mathbb{R}^n \times \mathbb{R}^m$ into $\mathbb{R}^n$. Fix $x_0 \in \mathbb{R}^n$ and $T > 0$. The *end-point mapping* is the mapping $E^{x_0,T} : u(.) \in \mathcal{U} \to x(T, x_0, u)$ where $\mathcal{U} \subset L^\infty[0, T]$ is the set of admissible controls such that the corresponding trajectory $x(., x_0, u)$ is defined on $[0, T]$. A control $u(.)$ and its corresponding trajectory are called *singular* on $[0, T]$ if $u(.) \in \mathcal{U}$ is such that the Fréchet derivative $E'^{x_0,T}$ of the end-point mapping is not of full rank n at $u(.)$.

Fréchet Derivative and Linearized System

Given a reference trajectory $x(.)$, $t \in [0, T]$, associated to $u(.)$ with $x(0) = x_0$, and solution of $\frac{dx}{dt}(t) = F(x(t), u(t))$, the system

$$\dot{\delta x}(t) = A(t)\delta x(t) + B(t)\delta u(t)$$

with

$$A(t) = \frac{\partial F}{\partial x}(x(t), u(t)), \ B(t) = \frac{\partial F}{\partial u}(x(t), u(t))$$

is called the *linearized system* along the control-trajectory pair $(u(.), x(.))$.

Let $M(t)$ be the fundamental matrix, $t \in [0, T]$ solution of

$$\dot{M}(t) = A(t)M(t), \qquad M(0) = I_n.$$

Integrating the linearized system with $\delta x(0) = 0$, one gets the following proposition.

Proposition 1 *The Fréchet derivative of $E^{x_0,T}$ at $u(.)$ is given by*

$$E_u'^{x_0,T}(v) = M(T) \int_0^T M^{-1}(t)B(t)v(t) dt.$$

S

Computation of the Singular Trajectories and Pontryagin Maximum Principle

According to the previous computations, a control $u(.)$ with corresponding trajectory $x(.)$ is singular on $[0, T]$ if the Fréchet derivative $E'^{x_0,T}$ is not of full rank at $u(.)$. This is equivalent to the condition that the linearized system is *not controllable* (Lee and Markus 1967).

Such a condition is difficult to verify directly since the linearized system is time-depending and the computation is associated to the Maximum Principle (Pontryagin et al. 1962).

Let p^* be a nonzero vector such that p^* is orthogonal to $\text{Im}(E'^{x_0,T})$ and let $p(t) = p^* M(T) M^{-1}(t)$; then $p(.)$ is solution of the *adjoint system*

$$\dot{p}(t) = -p(t)\frac{\partial F}{\partial u}(x(t), u(t))$$

and satisfies almost everywhere the equality

$$p(t)\frac{\partial F}{\partial u}(x(t), u(t)) = 0.$$

Introduce the *pseudo-Hamiltonian* $H(x, p, u) = \langle p, F(x, u) \rangle$, where $\langle ., . \rangle$ is the Euclidean inner product, one gets the following characterization.

Proposition 2 *If (x, u) is a singular control-trajectory pair on $[0, T]$, then there exists a nonzero adjoint vector $p(.)$ defined on $[0, T]$ such that (x, p, u) is solution a.e. of the following equations:*

$$\frac{dx}{dt} = \frac{\partial H}{\partial p}(x, p, u), \ \frac{dp}{dt} = -\frac{\partial H}{\partial x}(x, p, u)$$

$$\frac{\partial H}{\partial u}(x, p, u) = 0.$$

Application to the Lagrange Problem

Consider the problem

$$\frac{dx}{dt}(t) = F(x(t), u(t)), \min \int_0^T L(x(t), u(t)) dt$$

with $x(0) = x_0$, $x(T) = x_1$.

Introduce the *cost-extended pseudo-Hamiltonian*: $\tilde{H}(x, p, u) = \langle p, F(x, u) \rangle + p_0 L(x, u)$; it follows that the maximum principle is equivalent to the Lagrange multiplier rule presented in the introduction:

$$\frac{d\tilde{x}}{dt} = \frac{\partial \tilde{H}}{\partial \tilde{p}}(\tilde{x}, \tilde{p}, u), \frac{d\tilde{p}}{dt} = -\frac{\partial \tilde{H}}{\partial \tilde{x}}(\tilde{x}, \tilde{p}, u)$$

$$\frac{\partial \tilde{H}}{\partial u}(\tilde{x}, \tilde{p}, u) = 0$$

where $\tilde{x} = (x, x^0)$ is the extended state variable solution of $\frac{dx}{dt} = F(x, u), \frac{dx^0}{dt} = L(x, u)$ and $\tilde{p} = (p, p_0)$ is the extended adjoint vector. One has the condition $\langle \tilde{p}, \tilde{E}_u'^{x_0,T}(v) \rangle = 0$ where $\tilde{E}^{x_0,T}$ is the cost-extended end-point mapping.

The Role of Singular Extremals in Optimal Control

While the traditional treatment in optimization of singular extremals is to consider them as a pathology, in modern optimal control, they play an important role which is illustrated by two examples from *geometric optimal control*.

Singular Trajectories in Quantum Control

Up to a normalization (Lapert et al. 2010), the time minimization *saturation problem* is to steer in minimum time the magnetization vector $M = (x, y, z)$ from the north pole of the Bloch Ball $N = (0, 0, 1)$ to its center $O = (0, 0, 0)$. The evolution of the system is described by the *Bloch equation* in nuclear magnetic resonance (Levitt 2008)

$$\frac{dx}{dt} = -\Gamma x + u_2 z$$

$$\frac{dy}{dt} = -\Gamma - u_1 z$$

$$\frac{dz}{dt} = \gamma(1 - z) + u_1 y - u_2 x$$

where (Γ, γ) are proportional to the inverse of the relaxation times and $u = (u_1, u_2)$ is the control radio frequency-magnetic field bounded according to $|u| \leq M$. Due to the z-symmetry of revolution, one can restrict the problem to the 2D single-input case

$$\frac{dy}{dt} = -\Gamma y - uz, \frac{dz}{dt} = \gamma(1 - z) + uy$$

that can be written as $\frac{dq}{dt} = F(q) + uG(q)$.

According to the maximum principle, the time-optimal solutions are the concatenations of *regular extremals* for which $u(t) = M \operatorname{sign}\langle p(t), G(q(t)) \rangle$ and singular arcs where $\langle p(t), G(q(t)) \rangle = 0, \forall t$, and $p(t)$ is solution of the adjoint system. Differentiating with respect of time and using the *Lie bracket* notation $[X, Y](q) = \frac{\partial X}{\partial q}(q)Y(q) - \frac{\partial Y}{\partial q}(q)X(q)$, we get

$$\langle p, [G, F](q) \rangle = 0,$$

$$\langle p, [[G, F], G](q) \rangle + u \langle p, [[G, F], F](q) \rangle = 0.$$

This leads to two singular arcs:
- The vertical line $y = 0$, corresponding to the z-axis of revolution
- The horizontal line $z = \frac{\gamma}{2(\gamma - \Gamma)}$

The interesting physical case is when $2\Gamma > 3\gamma$ where the vertical singular line is such that $-1 < \frac{\gamma}{2(\gamma - \Gamma)} < 0$. In this case, the time minimum solution is represented on Fig. 1. On Fig. 2 we draw the experimental solution in the deoxygenated blood case, compared with the standard inversion recovery sequence.
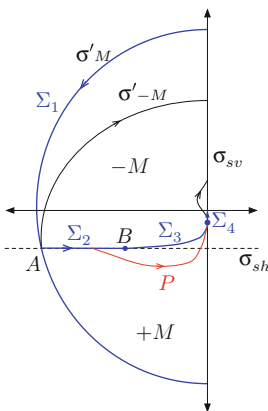
Singular Trajectories in Optimal Control, Fig. 1 The computed optimal solution is the following concatenation: bang arc σ'_M with the horizontal singular arc σ_{sh} followed by a bang arc P and finally the singular vertical arc σ_{sv}

Abnormal Extremals in SR Geometry

Sub-Riemannian geometry was introduced by R.W. Brockett as a generalization of Riemannian geometry (Brockett 1982; Montgomery 2002) with many applications in control (for instance, in motion planning (Bellaiche et al. 1998; Gauthier and Zakalyukin 2006) and quantum control). Its formulation in the framework of control theory is

$$\dot{q}(t) = \sum_{i=1}^{m} u_i(t) F_i(q(t)), \quad \min_{u(.)} \int_0^T (\sum_{i=1}^{m} u_i^2(t) dt)$$

where $q \in U$ open set in $\mathbb{R}^n$, $m < n$ and $F_1, \cdots, F_m$ are smooth vector fields which forms an orthonormal basis of the distribution they generate.

According to the maximum principle, normal extremals are solutions of the Hamiltonian vector field $\mathbf{H}_n$, $H_n = \frac{1}{2}(\sum_{i=1}^{m} H_i(q, p)^2)$, $H_i = \langle p, F_i(q) \rangle$ for $i = 1, \cdots m$. Again abnormal extremals can be computed by differentiating the constraint $H_i = 0$ along the extremals. Their first occurrence takes place in the so-called Martinet flat case: $n = 3, m = 2, F_1, F_2$ are given by

$$F_1 = \frac{\partial}{\partial x} + \frac{y^2}{2}\frac{\partial}{\partial z}, F_2 = \frac{\partial}{\partial y}$$

where $q = (x, y, z) \in U$ neighborhood of the origin, and the metric is given by $ds^2 = dx^2 + dy^2$. The singular trajectories are contained in the Martinet plane $M : y = 0$ and are the lines $z = z_0$. An easy computation shows that they are optimal for the problem. We represent below the role of the singular trajectories when computing the sphere of small radius, from the origin, intersected with the Martinet plane (Fig. 3).

Summary and Future Directions

Singular trajectories play an important role in many optimal control problem such as in quantum control and cancer therapy (Schättler and Ledzewicz 2012). They have to be carefully analyzed in any applications; in particular in

Singular Trajectories in Optimal Control, Fig. 2 Experimental result. Usual inversion sequence in *green*, optimal computed sequence in *blue*
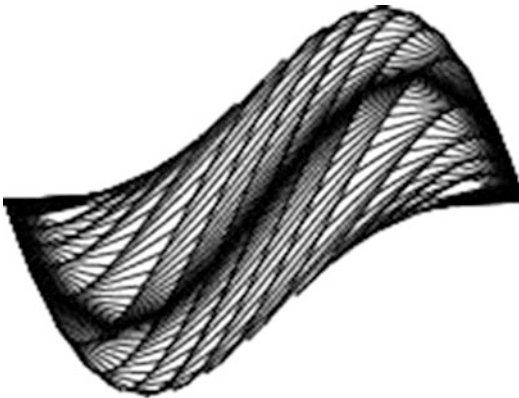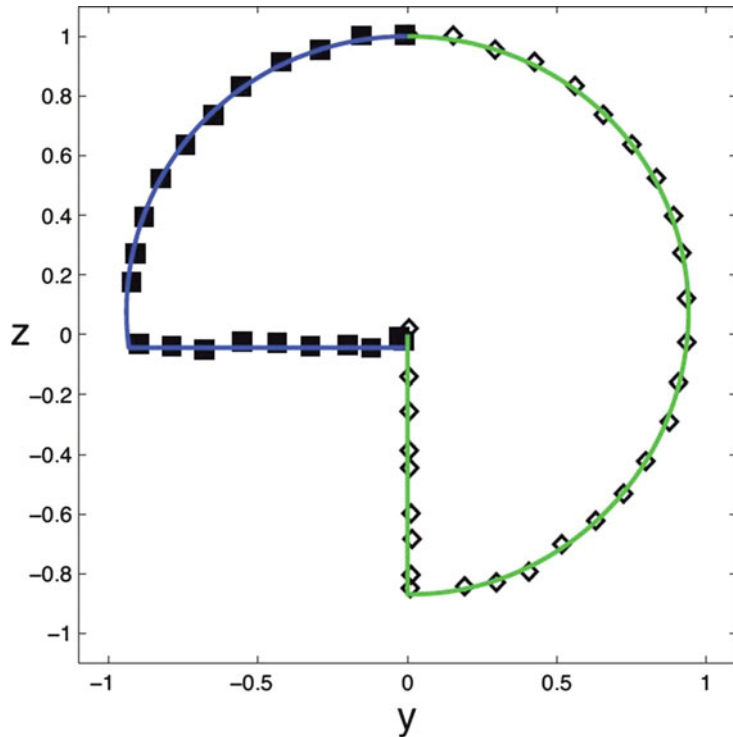
Singular Trajectories in Optimal Control, Fig. 3 Projection of the SR sphere on the xz-plane. The singular line is $x = t$ and the picture shows the pinching of the SR sphere in the singular direction

Boscain and Piccoli (2006) the authors provide for single-input systems in two dimensions a classification of optimal synthesis with singular arcs.

Additionally, from a theoretical point of view, singular trajectories can be used to compute feedback invariants for nonlinear systems (Bonnard and Chyba 2003). In relation, a purely mathemat-

ical problem is the classification of distributions describing the nonholonomic constraints in sub-Riemannian geometry (Montgomery 2002).

Cross-References

- ▶ Differential Geometric Methods in Nonlinear Control
- ▶ Feedback Stabilization of Nonlinear Systems
- ▶ Optimal Control and Pontryagin's Maximum Principle
- ▶ Robustness Issues in Quantum Control
- ▶ Sub-Riemannian Optimization

Bibliography

Agrachev A, Sarychev AV (1998) On abnormal extremals for lagrange variational problems. J Math Syst Estim Control 8(1):87–118

Agrachev A, Bonnard B, Chyba M, Kupka I (1997) Sub-Riemannian sphere in Martinet flat case. ESAIM Control Optim Calc Var 2:377–448

Bellaiche A, Jean F, Risler JJ (1998) Geometry of non-holonomic systems. In: Laumond JP (ed) Robot motion planning and control. Lecture notes in control and information sciences, vol 229. Springer, London, pp 55–91

Bliss G (1946) Lectures on the calculus of variations. University of Chicago Press, Chicago

Bloch A (2003) Nonholonomic mechanics and control. Interdisciplinary applied mathematics, vol 24. Springer, New York

Bonnard B, Chyba M (2003) Singular trajectories and their role in control theory. Mathématiques & applications, vol 40. Springer, Berlin

Bonnard B, Cots O, Glaser S, Lapert M, Sugny D, Zhang Y (2012) Geometric optimal control of the contrast imaging problem in nuclear magnetic resonance. IEEE Trans Autom Control 57(8):1957–1969

Boscain U, Piccoli B (2004) Optimal syntheses for control systems on 2-D manifolds. Mathématiques & applications, vol 43. Springer, Berlin

Brockett RW, (1982) Control theory and singular Riemannian geometry. New directions in applied mathematics. Springer, New York/Berlin, pp 11–27

Gauthier JP, Zakalyukin V (2006) On the motion planning problem, complexity, entropy, and nonholonomic interpolation. J Dyn Control Syst 12(3):371–404

Lapert M, Zhang Y, Braun M, Glaser SJ, Sugny D (2010) Singular extremals for the time-optimal control of dissipative spin 1/2 particles. Phys Rev Lett 104:083001

Lapert M, Zhang Y, Janich M, Glaser SJ, Sugny D (2012) Exploring the physical limits of saturation contrast in magnetic resonance imaging. Nat Sci Rep 2:589

Lee EB, Markus L (1967) Foundations of optimal control theory. Wiley, New York/London/ Sydney

Levitt MH (2008) Spin dynamics: basics of nuclear magnetic resonance, 2nd edn. Wiley, Chichester/Hoboken

Montgomery R (2002) A tour of subriemannian geometries, their geodesics and applications. Mathematical surveys and monographs, vol 91. American Mathematical Society, Providence

Schättler H, Ledzewicz U (2012) Geometric optimal control: theory, methods and examples. Interdisciplinary applied mathematics, vol 38. Springer, New York

Pontryagin LS, Boltyanskii VG, Gamkrelidze RV, Mishchenko EF (1962) The mathematical theory of optimal processes (Translated from the Russian by Trirogoff KN; edited by Neustadt LW). Wiley Interscience, New York/London

Small Signal Stability in Electric Power Systems

Vijay Vittal
Arizona State University, Tempe, AZ, USA

Abstract

Small signal rotor angle stability analysis in power systems is associated with insufficient damping of oscillations under small disturbances.

Rotor angle oscillations due to insufficient damping have been observed in many power systems around the world. This entry overviews the predominant approach to examine small signal rotor angle stability in large power systems using eigenvalue analysis.

Keywords

Small Signal Rotor Angle Stability in Power Systems

As power system interconnections grew in number and size, automatic controls such as voltage regulators played critical roles in enhancing reliability by increasing the synchronizing capability between the interconnected systems. As technology evolved the capabilities of voltage regulators to provide synchronizing torque following disturbances were significantly enhanced. It was, however, observed that voltage regulators tended to reduce damping torque, as a result of which the system was susceptible to rotor angle oscillatory instability. An excellent exposition of the mechanism and the underlying analysis is provided in the textbooks (Anderson and Fouad 2003; Sauer and Pai 1998; Kundur 1993), and a number of practical aspects of the analysis are detailed in Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000). Two types of rotor angle oscillations are commonly observed. Low-frequency oscillations involving synchronous machines in different operating areas are commonly referred to as inter-area oscillations. These oscillations are typically in the 0.1–2 Hz frequency range. Oscillations between local machines or a group of machines at a power plant are referred to as plant mode oscillations. These oscillations are typically above the 2 Hz frequency range. The modes associated with rotor angle oscillations are also termed inertial modes of oscillation. Other modes of oscillations associated with the various

S

controls also exist. With the integration of significant new wind and photovoltaic generation which are interconnected to the grid using converters, new modes of oscillation involving the converter controls and conventional synchronous generator states are being observed.

The basis for small signal rotor angle stability analysis is that the disturbances considered are small enough to justify the use of linear analysis to examine stability (Kundur et al. 2004). As a result, Lyapunov's first method Vidyasagar (1993) provides the analytical underpinning to analyze small signal stability. Eigenvalue analysis is the predominant approach to analyze small signal rotor angle stability in power systems. Commercial software packages that utilize sophisticated algorithms to analyze large-scale power systems with the ability to handle detailed models of power system components exist.

The power system representation is described by a set of nonlinear differential algebraic equations shown in (1)

$$\dot{x} = f(x, z)$$
$$0 = g(x, z) \tag{1}$$

where x is the state vector and z is a vector of algebraic variables. Small signal stability analysis involves the linearization of (1) around a system operating point which is typically determined by conducting a power flow analysis:

$$\begin{bmatrix} \Delta \dot{x} \\ 0 \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} \tag{2}$$

The power system state matrix can be obtained by eliminating the vector of algebraic variables Δz in (2)

$$\Delta \dot{x} = \left(J_1 - J_2 J_4^{-1} J_3 \right) \Delta x = A \Delta x \tag{3}$$

where A represents the system state matrix. Based on Lyapunov's first method, the eigenvalues of A characterize the small signal stability behavior of the nonlinear system in a neighborhood of the operating point around which the system is linearized. The eigenvectors corresponding to the eigenvalues also provide

significant qualitative information. For each eigenvalue λ_i, there exists a vector u_i known as the right eigenvector of A which satisfies the equation

$$A u_i = \lambda_i u_i \tag{4}$$

There also exists a row vector v_i known as the left eigenvector of A which satisfies

$$v_i A = \lambda_i v_i \tag{5}$$

For a system which has distinct eigenvalues, the right and left eigenvectors form an orthogonal set governed by

$$\begin{aligned} v_i u_j &= k_{ij} \\ \text{where} \\ k_{ij} &\neq 0 \ i = j \\ k_{ij} &= 0 \ i \neq j \end{aligned} \tag{6}$$

One set (either right or left) of eigenvectors are usually scaled to unity and the other set obtained by solving (6) with $k_{ij} = 1$. The right eigenvectors can be assembled together as columns of a square matrix U, and the corresponding left eigenvectors can be assembled as rows of a matrix V; then

$$V = U^{-1} \tag{7}$$

and

$$VAU = \Lambda \tag{8}$$

where Λ is a diagonal matrix with the distinct eigenvalues as the diagonal entries. The relationship in (8) is a similarity transformation and in the case of distinct eigenvalues provides a pathway to obtain solutions to the linear system of equations (3). Applying the following similarity transformation to (3)

$$\Delta x = U z \rightarrow \Delta x_i(t) = \sum_{j=1}^{n} u_{ij} z_j e^{\lambda_j t} \tag{9}$$

$$U \dot{z} = A U z \tag{10}$$

$$\dot{z} = U^{-1} A U z = V A U z = \Lambda z \tag{11}$$

$$\dot{z}_i(t) = \lambda_i z_i \Rightarrow z_i(t) = z_i(0) e^{\lambda_i t} \tag{12}$$

$$z_i(0) = v_i^T \Delta x(0) \tag{13}$$

$$z_i\,(t) = v_i^T \,\Delta x\,(0)\,e^{\lambda_i t} \qquad (14)$$

From (9) and (14), it can be observed that the right eigenvector describes how each mode of the system is distributed throughout the state vector (and is referred to as the mode shape), and the left eigenvector in conjunction with the initial conditions of the system state vector determines the magnitude of the mode. The right eigenvector or the mode shape has been often used to identify dynamic patterns in small signal dynamics. One problem with the mode shape is that it is dependent on the units and scaling of the state variables as a result of which it is difficult to compare the magnitudes of entries that are disparate and correspond to states that impact the dynamics differently. This resulted in the development of the participation factors (Pérez-Arriaga et al. 1982) which are dimensionless and independent of the choice of units. The participation factor is expressed as

$$p_{ik} = v_{ik} u_{ik} \qquad (15)$$

The magnitude of the participation factor measures the relative participation of the ith state variable in the kth mode and vice versa.

Small Signal Stability Analysis Tools for Large Power Systems

Efficient software tools exist that facilitate the application of the methods in section "Small Signal Rotor Angle Stability in Power Systems" to large power systems (Powertech 2012; Martins 1989). These tools incorporate detailed models of power system components and also leverage the sparsity in power systems. The building of the A matrix is a complex task for large power systems with a multitude of dynamic components. The approach in Powertech (2012) utilizes a technique where state space equations are developed for each dynamic component in the system using a solved power flow solution and the dynamic data description for a given system. These state space equations are then coupled based on the system topology, and the system A matrix is derived as in (3). Reference Martins (1989) takes advan-

tage of the sparsity of the Jacobian matrix in (2) and develops efficient algorithms to determine the eigenvalues and eigenvectors. The software tools also provide the flexibility of a number of different options with regard to eigenvalue computations:

1. Calculation of a specific eigenvalue at a specified frequency or with a specified damping ratio
2. Simultaneous calculation of a group of relevant eigenvalues in a specified frequency range or in specified damping ratio range

In addition to the features described above, commercial software packages also provide features to evaluate:

1. Frequency response plots
2. Participation factors
3. Transfer functions, residues, controllability, and observability factors
4. Linear time response to step changes
5. Eigenvalue sensitivities to changes in specified parameters

Applications of Small Signal Stability Analysis in Power Systems

Small signal stability analysis tools are used for a range of applications in power systems. These applications include:

Analysis of local stability problems – These types of stability problems are primarily associated with the tuning of control associated with the synchronous generator, converter interconnected renewable resources, and HVDC link current control. In certain cases analysis of local stability problems could also involve design of supplementary controllers which enhance the stability region. Since the stability problem pertains to a local portion of the power system, there is significant flexibility in modeling the system. In many instances local stability problems facilitate the use of a simple representation of a power system which could include the particular machine or a local group of machines in question together with a highly equivalenced representation of the rest of the system. In cases where controls other than generator controls influence stability, e.g.,

S

static VAr compensators or HVDC links, the system representation would need to be extended to include portions of the system where these devices are located. Typical small signal stability problems that are analyzed include:

1. Power system stabilizer design
2. Automatic voltage regulator tuning
3. Governor tuning
4. DC link current control
5. Small signal stability analysis for subsynchronous resonance
6. Load modeling effects on small signal stability

References Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000) provide comprehensive examples of the analysis conducted for each of the problems listed above.

Analysis of global stability problems – These types of stability problems are associated with controls that impact generators located in different areas of the power systems. The analysis of these inter-area problems requires a more systematic approach and involves representation of the power system in greater detail. The problems that are analyzed under this category include:

1. Power system stabilizer design
2. HVDC link modulation
3. Static VAr compensator controls

References Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) and Rogers (2000) again provide details of the analysis conducted for each of the problems listed under this category.

Cross-References

▶ Lyapunov Methods in Power System Stability
▶ Lyapunov's Stability Theory
▶ Power System Voltage Stability
▶ Stability: Lyapunov, Linear Systems

Bibliography

Anderson PM, Fouad AA (2003) Power system control and stability, 2nd edn. Wiley Interscience, Hoboken

Eigenanalysis and Frequency Domain Methods for System Dynamic Performance (1989) IEEE Special Publication, 90TH0292-3-PWR

Kundur P (1993) Power system stability and control. McGraw Hill, San Francisco

Kundur P, Paserba J, Ajjarapu V, Andersson G, Bose A, Canizares C, Hatziargyriou N, Hill D, Stankovic A, Taylor C, Van Cutsem T, Vittal V (2004) Definition and classification of power system stability. IEEE/CIGRE joint task force on stability terms and definitions report. IEEE Trans Power Syst 19:1387–1401

Martins N (1989) Efficient eigenvalue and frequency response methods applied to power system small signal stability studies. IEEE Trans Power Syst 1:74–82

Pérez-Arriaga IJ, Verghese GC, Schweppe FC (1982) Selective modal analysis with applications to electric power systems, part 1: Heuristic introduction. IEEE Trans Power Appar Syst 101:3117–3125

Powertech (2012) Small signal analysis tool (SSAT) user manual. Powertech Labs Inc, Surrey

Rogers G (2000) Power system oscillations. Kluwer Academic, Dordrecht

Sauer PW, Pai MA (1998) Power system dynamics and stability. Prentice Hall, Upper Saddle River

Vidyasagar M (1993) Nonlinear systems analysis, 2nd edn. Prentice Hall, Englewood Cliffs

Spatial Description of Biochemical Networks

Pablo A. Iglesias
Electrical & Computer Engineering, The Johns Hopkins University, Baltimore, MD, USA

Abstract

Many biological behaviors require that biochemical species be distributed spatially throughout the cell or across a number of cells. To explain these situations accurately requires a spatial description of the underlying network. At the continuum level, this is usually done using reaction-diffusion equations. Here we demonstrate how this class of models arises. We also show how the framework is used in two popular models proposed to explain spatial patterns during development.

Keywords

Diffusion; Morphogen gradient; Pattern formation; Reaction-diffusion; Turing instability

Introduction

Cells are complex environments consisting of spatially segregated entities, including the nucleus and various other organelles. Even within these compartments, the concentrations of various biochemical species are not homogeneous, but can vary significantly. The proper localization of proteins and other biochemical species to their respective sites is important for proper cell function. This can be because the spatial distribution of signaling molecules itself confers information, such as when a cell needs to respond to a spatially graded cue to guide its motion (Iglesias and Devreotes 2008) or growth pattern (Lander 2013). Alternatively, information that is obtained in one part of the cell must be transmitted to another part of the cell, as when receptor-ligand binding at the cell surface leads to transcriptional responses in the nucleus. Frequently, describing the action of a biological network accurately requires not only that one account for the chemical interactions between the different components but that the spatial distribution of the signaling molecules also be considered.

Accounting for Spatial Distribution in Models

Mathematical models of biological networks usually assume that reactions take place in well-stirred vessels in which the concentrations of the interacting species are spatially homogeneous and hence need not be accounted for explicitly. These systems also assume that the volume is constant. When the spatial location of molecules in cells is important, the concentration of species changes in both time and space.

Compartmental Models
One way to account for spatial distribution of signaling components is through compartmental models. As the name suggests, in these models the cell is divided into different regions that are segregated by membranes. Within each compartment, the concentration of the network species

is assumed to be spatially homogeneous. The membranes in these models can be assumed to be either permeable or impermeable. In permeable membranes, information passes through small openings, such as ion channels or nuclear pores, which allow molecules to move from one side of the membrane to the other. With impermeable membranes, information must be transduced by transmembrane signaling elements, such as cell-surface receptors, that bind to a signaling molecule in one side of the membrane and release a secondary effector on the other side. Note that in this case, the membrane itself acts as a third compartment.

Compartmental models offer simplicity, since the reactions that happen in a single region obey the same reaction kinetics usually assumed in spatially homogeneous models. Even when the reactions involve more than one compartment, as in ligand-receptor binding, this can still be described by the usual reaction dynamics. Care must be taken, however, to account properly for the different effects on the respective concentrations as molecules move from one compartment to another. In models of spatially homogeneous systems, there is little practical difference between writing the ordinary-differential equations in terms of molecule numbers or concentrations, since the two are proportional to each other according to the volume, which is constant. In a compartmental model, if the molecule moves from one compartment to another, there is conservation of molecule numbers, but not concentrations. For example, if a species is found in two compartments with volumes V_1 and V_2 and transfer rates k_{12} and k_{21} s^{-1}, then the differential equations describing transport between compartments can be expressed in terms of numbers (n_1 and n_2) as follows:

$$\frac{dn_1}{dt} = -k_{12}n_1 + k_{21}n_2$$
$$\frac{dn_2}{dt} = +k_{12}n_1 - k_{21}n_2.$$

Dividing by the respective volumes ($C_1 = n_1/V_1$ and $C_2 = n_2/V_2$), we obtain equations for the concentrations

$$\frac{dC_1}{dt} = -k_{12}C_1 + k_{21}(\tfrac{V_2}{V_1})C_2$$

$$\frac{dC_2}{dt} = +k_{12}(\tfrac{V_1}{V_2})C_1 - k_{21}C_2.$$

In the former case, the two equations add to zero, indicating that $n_1(t) + n_2(t) =$ constant. In the latter, if $V_1 \neq V_2$, then $C_1(t) + C_2(t)$ varies over time as molecules move from one compartment to the other.

Diffusion and Advection

If the distribution of molecules inside any single compartment is spatially heterogeneous, then models must account for this spatial distribution. At the continuum level, this is done using reaction-diffusion equations. The basic assumption is a conservation principle expressed as a continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla j = f,$$

which relates the changes in the density (ρ) of a conserved quantity (in our case, the concentration of a species: $\rho = C$) to the flux j and any net production f. In biological networks, the latter represents the net effect of all the reactions that affect the concentration of the species including binding, unbinding, production, degradation, post-translational modifications, etc.

In biological models, the flux term usually comes from one of two sources: diffusion or advection. According to Fick's law, diffusive flux is proportional to the negative gradient of the concentration of the species as particles move from regions of high concentration to regions of low concentration. The coefficient of proportionality is the diffusion coefficient, D:

$$j_{\text{diff}} = -D\nabla C.$$

Fick's law describes thermally driven Brownian motion of molecules at the continuum level. If the species is embedded in a moving field, then the flux is proportional to the velocity of the underlying fluid. In this case, we have advective flow:

$$j_{\text{adv}} = vC.$$

In biological systems, advection can arise because of the movement of the cytoplasm, but it can also represent directed transport of molecules, such as the movement of cargo along filaments by processive motors. In general, molecules exhibit both diffusive and advective motion: $j = j_{\text{diff}} + j_{\text{adv}}$, leading to

$$\frac{\partial C}{\partial t} + \nabla(-D\nabla C + vC) = f,$$

which, under the assumption that the diffusion coefficient and the transport velocity are independent of spatial location, leads to the reaction-diffusion-advection equation:

$$\frac{\partial C}{\partial t} = D\nabla^2 C - v\nabla C + f.$$

Being a second-order partial differential, the solution requires an initial condition and two boundary conditions. Common choices for the latter include periodic (e.g., in models of closed boundaries) or no-flux (to describe the impermeability of membranes) assumptions.

Measuring Diffusion Coefficients

Invariably, solving the reaction-diffusion equation requires knowledge of the diffusion coefficient of the molecule. Experimentally, this can be done in a number of ways. In fluorescence recovery after photobleaching (FRAP), a laser is used to photobleach normally fluorescent molecules in a specific area of the cell. As these "dark" molecules are replaced by fluorescent molecules from non-bleached areas, the fluorescent intensity of the bleached area recovers. Higher diffusion leads to faster recovery. The time to half recovery, $\tau_{1/2}$, can be used to estimate D. If recovery occurs by lateral diffusion, then

$$D = \frac{r_0^2 \gamma}{4\tau_{1/2}}$$

where r_0 is the $1/e^2$ radius of the Gaussian profile laser beam and γ is a parameter that depends on the extent of photobleaching, which ranges from 1 to 1.2 (Chen et al. 2006).

These days, it is increasingly common to measure lateral diffusion coefficients by observing the trajectory of single molecules. A molecule with diffusion coefficient D undergoing Brownian motion in a two-dimensional environment is expected to have mean-square displacement (MSD) equal to

$$\langle r^2 \rangle = 4Dt.$$

Thus, the coefficient D can be obtained by measuring how the MSD changes as a function of the time interval t. This method can also show if the molecule is undergoing advection in which case

$$\langle r^2 \rangle = 4Dt + v^2 t^2.$$

This super-diffusive behavior can be seen in the concave nature of the plot of $\langle r^2 \rangle$ against t. This plot will also reveal barriers to diffusion. For example, if the molecule is confined to move in a circular region of radius a, then, as t increases, $\langle r^2 \rangle$ cannot exceed a^2.

Both these methods work best for molecules diffusing on a membrane. For molecules diffusing in the cytoplasm, the three-dimensional imaging required is considerably more difficult, particularly since the diffusion of particles in the cytoplasm ($D \sim 1\text{--}10\,\mu\text{m}^2\,\text{s}^{-1}$) is usually orders of magnitude greater than for membrane-bound proteins ($D \sim 0.01\text{--}0.1\,\mu\text{m}^2\,\text{s}^{-1}$). In this case, an analytical expression can be used to estimate the diffusion coefficient. The diffusion coefficient of a spherical particle of radius r moving in a low Reynolds number liquid with viscosity η is given by the Stokes-Einstein equation:

$$D = \frac{k_B T}{6\pi \eta r}.$$

The exact viscosity of the cell is unknown, but estimates that η is approximately five times that of water lead to diffusion coefficients of cytoplasmic proteins that match those measured using FRAP.

Diffusion-Limited Reaction Rates

Even in compartments that are considered well stirred, the diffusion of molecules is necessary for reactions to take place. In particular, before two molecules can react, they must come together. To see how diffusion influences this, suppose that spherical molecules of species A and B with radii r_A and r_B, respectively, come together to form a complex AB at a rate k_d. This rate represents the likelihood that molecules of A and B collide at random and hence will depend on the diffusion properties of the two species. The molecules in this complex can dissociate at rate k_d' or can be converted to species C at rate k_r. Thus, the overall reaction involves two steps:

$$A + B \underset{k_d'}{\overset{k_d}{\rightleftharpoons}} AB$$

$$AB \xrightarrow{k_r} C.$$
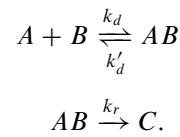
Assuming that the system is at quasi-steady-state, that is, the concentration of AB is constant, the effective rate of production C is given by

$$k_{\text{eff}} = \frac{k_d k_r}{k_d' + k_r}.$$

There are two regions of operation. If $k_d' \gg k_r$, then $k_{\text{eff}} \approx k_r(k_d/k_d')$. In this case production is said to be reaction limited. If $k_d' \ll k_r$, then $k_{\text{eff}} \approx k_d$ and production is diffusion limited. In this case, it is possible to find k_d as a function of the species' diffusion coefficients.

Assume that species A is stationary, in which case the effective diffusion is the sum of the two diffusion coefficients: $D = D_A + D_B$. The concentration of species B depends on the distance away from molecules of A. Because we assume that the reaction rate is fast, at the point of contact ($r^\star = r_A + r_B$) the concentration is zero since any molecules of AB are quickly converted to C. At the other extreme, as $r \to \infty$, the concentration approaches the bulk concentration B_0. According to Fick's law, this concentration gradient causes a flux density given by $j = -D(\partial B/\partial r)$. The total flux into a sphere of radius r is then

$$J = 4\pi r^2 j = -4\pi D r^2 \frac{\partial B}{\partial r},$$

which, at steady state, is constant. Solving this equation for $B(r)$ using the two boundary equations leads to a flux

$$J = -4\pi D B_0 r^\star,$$

from which we have that

$$k_d = 4\pi D r^\star.$$

A typical value for k_d, using the Einstein-Stokes formula, is

$$4\pi \left(2 \times \frac{k_B T}{6\pi \eta (r^\star/2)} \right) r^\star = \frac{8 k_B T}{3\eta}$$
$$\approx 10^3 \, \mu\mathrm{m}^{-1} \, \mathrm{s}^{-1}.$$

Spatial Patterns

The effect of spatial heterogeneities has been of long interest to developmental biologists, who study how spatial patterns arise. Two distinct models have been proposed to explain how this patterning can arise. Here we introduce these models and discuss their relative merits. Though usually seen as competing models, there is recent evidence suggesting that both models may play complementary roles during development (Reth et al. 2012).

Morphogen Gradients

A morphogen is a diffusible molecule that is produced or secreted at one end of an organism. Diffusion away from the localized source forms a concentration gradient along the spatial dimension. Morphogens are used to control gene expression of cells lying along this spatial domain. Thus, a morphogen gradient gives rise to spatially dependent expression profiles that can account for spatial developmental patterns (Rogers and Schier 2011).

The mathematics behind the formation of a morphogen gradient are relatively straightforward. The concentration of the morphogen is denoted by $C(x, t)$. There is a constant flux (j_0) at one end ($x = 0$) of a finite one-dimensional

domain of length L, but the morphogen cannot exit at the other end. The species diffuses inside the domain and also decays at a rate proportional to its concentration ($f = -kC$). Thus, the concentration is governed by the reaction-diffusion equation:

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - kC,$$

with boundary conditions: $D \frac{\partial C}{\partial x} = -j_0$ at $x = 0$, and $D \frac{\partial C}{\partial x} = 0$ at $x = L$. We focus on the steady state:

$$\frac{\mathrm{d}^2 \bar{C}}{\mathrm{d}x^2} = \frac{k}{D} \bar{C},$$

so that the initial condition is not important. In this case, the distribution of the species is given by

$$\bar{C}(x) = \frac{\lambda j_0}{D} \frac{\cosh([L - x]/\lambda)}{\sinh(L/\lambda)}.$$

Thus, the shape of the gradient is roughly exponential with parameter $\lambda = \sqrt{D/k}$, known as the dispersion, which specifies the average distance that molecules diffuse into the domain before they are degraded or inactivated. Equally important in determining the gradient, however, is the spatial dimension (L) relative to the dispersion, $\Phi = L/\lambda$, a ratio known as the Thiele modulus. If $\Phi \ll 1$, then the concentration will be approximately homogeneous. Alternatively, $\Phi \gg 1$ leads to a sharp transition close to the boundary where there is flux and a relatively flat concentration thereafter.

Though morphogen gradients are commonly used to describe signaling during development, where the gradient can extend across a number of cells, the mathematics described above are equally suitable for describing concentration gradients of intracellular proteins. In this case, the dimension of the cell has a significant effect on the shape of the gradient (Meyers et al. 2006).

As discussed above, morphogen gradients are established in an open-loop mode. As such, the actual concentration experienced at a point downstream of the source of the morphogen will vary depending on a number of parameters, including the flux j_0 and the rate of degradation k.

Moreover, because the concentration of the morphogen decreases as the distance from the source grows, the relative stochastic fluctuations will increase. How to manage this uncertainty is an active area of research (Rogers and Schier 2011; Lander 2013).

Diffusion-Driven Instabilities

In 1952, Alan Turing proposed a model of how patterns could arise in biological systems (Turing 1952). His interest was in explaining how an embryo, initially spherical, could give rise to a highly asymmetric organism. He posited that the breaking of symmetry could be a result of the change in the stability of the homogeneous state of the network which would amplify small fluctuations inherent in the initial symmetry. Turing sought to explain how these instabilities could arise using only reaction-diffusion systems.

To illustrate how diffusion-driven instabilities can arise, we work with a single two-species linear reaction network:

$$\frac{\partial}{\partial t} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = A \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \frac{\partial^2}{\partial x^2} D \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

where $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ specifies the reaction terms and the diagonal matrix $D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$ the diffusion coefficients.

We assume that, in the absence of diffusion, the system is stable, so that $\det(A) > 0$ and $\text{trace}(A) < 0$. When considering diffusion in a one-dimensional environment of length L, we must consider the spatial modes, which are of the form $\exp(iqx)$. In this case, stability of the system requires that $\text{trace}(A - q^2 D) < 0$ and $\det(A - q^2 D) > 0$. The former is always true, since $\text{trace}(A - q^2 D) = \text{trace}(A) - q^2(D_1 + D_2) < \text{trace}(A) < 0$. However, the condition on the determinant can fail since

$$\det(A - q^2 D) = D_1 D_2 q^4 - q^2(a_{22}D_1 + a_{11}D_2)$$
$$+ \det(A). \tag{1}$$

Since $\det(A) > 0$, diffusion-driven instabilities can only occur if the term $a_{22}D_1 + a_{11}D_2 > 0$, by which it follows that at least one of a_{11} or a_{22}

must be positive. Since $\text{trace}\, A < 0$, it follows that the diagonal terms must have opposite sign. Usually, it is assumed that $a_{11} > 0$ and that $a_{22} < 0$. Since $\det(A) > 0$, it follows that a_{12} and a_{21} must also have opposite sign.

These requirements in the sign pattern of the two molecules lead to one of two classes of systems. In the first class, known as activator/inhibitor systems, the activator (assume species 1) is autocatalytic ($a_{11} > 0$) and also stimulates the inhibitor ($a_{21} > 0$), which negatively regulates the activator ($a_{12} < 0$). In the other class, known as substrate-depletion systems, a product (species 1) is autocatalytic ($a_{11} > 0$), but in its production consumes ($a_{21} < 0$) the substrate (species 2) whose presence is needed for formation of the product ($a_{12} > 0$). Note that both systems involve an autocatalytic positive feedback loop ($a_{11} > 0$), as well as a negative feedback loop involving both species ($a_{12}a_{21} < 0$).

The stability condition also imposes a necessary condition on the dispersion of the two species, ($\lambda_i = \sqrt{D_i/|a_{ii}|}$), since

$$a_{22}D_1 + a_{11}D_2 > 0 \quad \Longrightarrow \quad -\lambda_1^2 + \lambda_2^2 > 0$$

Thus, the species providing the negative feedback (inhibitor or substrate) must have higher dispersion ($\lambda_2 > \lambda_1$). This requirement is usually referred to as local activation and long-range inhibition.

These conditions are necessary, but not sufficient. They ensure that the parabola defined by Eq. 1 has real roots. However, when diffusion takes place in finite domains, the parameter q can only take discrete values $q = 2\pi n/L$ for integers n. Thus, for a spatial mode to be unstable, it must be that $\det(A - q^2 D) < 0$ at specific values of q corresponding to integers n. If the dimension of the domain is changing, as would be expected in a growing domain, the parameter q^2 will decrease over time suggesting that higher modes may lose stability. Thus, the nature of the pattern may evolve over time.

Over the years, Turing's framework has been a popular model among theoretical biologists and has been used to explain countless patterns

seen in biological systems. It has not had the same level of acceptance among biologists, likely because of the difficulty of mapping a complex biological system involving numerous interacting species into the simple nature of the theoretical model (Kondo and Miura 2010).

Summary and Future Directions

Spatial aspects of biochemical signaling are increasingly playing a role in the study of cellular signaling systems. Part of this interest is the desire to explain spatial patterns seen in sub-cellular localizations observed through live cell imaging using fluorescently tagged proteins. The ever-increasing computational power available for simulations is also facilitating this progress. Specially built spatial simulation software, such as the Virtual Cell, is freely available and tailor-made for biological simulations enabling simulation of spatially varying reaction networks in cells of varying size and shape (Cowan et al. 2012).

Of course, cell shapes are not static, but evolve in large part due to the effect of the underlying biochemical system. This requires simulation environments that solve reaction-diffusion systems in changing morphologies. This has received considerable interest in modeling cell motility (Holmes and Edelstein-Keshet 2013).

Another aspect of spatial models that is only now being addressed is the role of mechanics in driving spatially dependent models. For example, it has recently been shown that the interaction between biochemistry and biomechanics can itself drive Turing-like instabilities (Goehring and Grill 2013).

Finally, we note that our discussion of spatially heterogeneous signaling has been based on continuum models. As with spatially invariant systems, this approach is only valid if the number of molecules is sufficiently large that the stochastic nature of the chemical reactions can be ignored. In fact, spatial heterogeneities may lead to localized spots requiring a stochastic approach, even though the molecule numbers are such that a continuum approach would be acceptable if the

cell were spatially homogeneous. The analysis of stochastic interactions in these systems is still much in its infancy and is likely to be an increasingly important area of research (Mahmutovic et al. 2012).

Cross-References

▶ Deterministic Description of Biochemical Networks
▶ Monotone Systems in Biology
▶ Robustness Analysis of Biological Models
▶ Stochastic Description of Biochemical Networks

Bibliography

Chen Y, Lagerholm BC, Yang B, Jacobson K (2006) Methods to measure the lateral diffusion of membrane lipids and proteins. Methods 39:147–153

Cowan AE, Moraru II, Schaff JC, Slepchenko BM, Loew LM (2012) Spatial modeling of cell signaling networks. Methods Cell Biol 110:195–221

Goehring NW, Grill SW (2013) Cell polarity: mechanochemical patterning. Trends Cell Biol 23: 72–80

Holmes WR, Edelstein-Keshet L (2013) A comparison of computational models for eukaryotic cell shape and motility. PLoS Comput Biol 8:e1002793; 2012

Iglesias PA, Devreotes PN (2008) Navigating through models of chemotaxis. Curr Opin Cell Biol 20: 35–40

Kondo S, Miura T (2010) Reaction-diffusion model as a framework for understanding biological pattern formation. Science 329:1616–1620

Lander AD (2013) How cells know where they are. Science 339:923–927

Mahmutovic A, Fange D, Berg OG, Elf J (2012) Lost in presumption: stochastic reactions in spatial models. Nat Methods 9:1163–1166

Meyers J, Craig J, Odde DJ (2006) Potential for control of signaling pathways via cell size and shape. Curr Biol 16:1685–1693

Rogers KW, Schier AF (2011) Morphogen gradients: from generation to interpretation. Annu Rev Cell Dev Biol 27:377–407

Sheth R, Marcon L, Bastida MF, Junco M, Quintana L, Dahn R, Kmita M, Sharpe J, Ros MA (2012) Hox genes regulate digit patterning by controlling the wavelength of a turing-type mechanism. Science 338: 1476–1480

Turing AM (1952) The chemical basis of morphogenesis. Philos Trans R Soc Lond 237:37–72

Spectral Factorization

Michael Sebek
Department of Control Engineering, Faculty of
Electrical Engineering, Czech Technical
University in Prague, Prague 6, Czech Republic

Abstract

For more than half a century, spectral factorization is encountered in various fields of science and engineering. It is a useful tool in robust and optimal control and filtering and many other areas. It is also a nice control-theoretical concept closely related to Riccati equation. As a quadratic equation in polynomials, it is a challenging algebraic task.

Keywords

Controller design; H_2-optimal control; H_∞-optimal control; J-spectral factorization; Linear systems; Polynomial; Polynomial equation; Polynomial matrix; Polynomial methods; Spectral factorization

Polynomial Spectral Factorization

As a mathematical tool, the spectral factorization was invented by Wiener in 1940s to find a frequency domain solution of optimal filtering problems. Since then, this technique has turned up numberless applications in system, network and communication theory, robust and optimal control, filtration, prediction and state reconstruction. Spectral factorization of scalar polynomials is naturally encountered in the area of single-input single-output systems.

In the context of continuous-time problems, real polynomials in a single complex variable s are typically used. For such a polynomial $p(s)$, its *adjoint* $p^*(s)$ is defined by

$$p^*(s) = p(-s), \tag{1}$$

which results in flipping all roots across the imaginary axis. If the polynomial is *symmetric,* then $p^*(s) = p(s)$ and its roots are symmetrically placed about the imaginary axis.

The *symmetric spectral factorization* problem is now formulated as follows: Given a symmetric polynomial $b(s)$,

$$b^*(s) = b(s), \tag{2}$$

that is also positive on the imaginary axis

$$b(i\omega) > 0 \quad \text{for all real } \omega, \tag{3}$$

find a real polynomial $x(s)$, which satisfies

$$x(s)x^*(s) = b(s) \tag{4}$$

as well as

$$x(s) \neq 0, \quad Res \geq 0. \tag{5}$$

Such an $x(s)$ is then called a *spectral factor* of $b(s)$. By (5), the spectral factor is a stable polynomial in the continuous-time (Hurwitz) sense.

Obviously, (4) is a quadratic equation in polynomials and its stable solution is the desired spectral factor.

Example 1 Given

$$b(s) = 4 + s^4 = (1 + j + s)(1 - j + s)$$
$$(1 + j - s)(1 - j - s),$$

(4) results in the spectral factor

$$x(s) = 2 + 2s + s^2 = (1 + j + s)(1 - j + s).$$

When the right-hand side polynomial $b(s)$ has some imaginary-axis roots, the problem formulated strictly as above becomes unsolvable since (3) does not hold and hence (5) cannot be fulfilled. A more relaxed formulation may then find its use requiring only $b(i\omega) \geq 0$ instead of (3) and $x(s) \neq 0$ only for $Res > 0$ instead of (5). Clearly, the imaginary-axis roots of $b(s)$ must then appear in $x(s)$ and $x^*(s)$ as well.

S

In the realm of discrete-time problems, one usually encounters *two-sided polynomial*s, which are polynomial-like objects (In fact, one can stay with standard one-sided polynomials (either in nonnegative or in nonpositive powers only), if every adjoint $p^*(z)$ is multiplied by proper power of z to create a one-sided polynomial $\bar{p}(z) = p^*(z)z^n$.) with positive and/or negative powers of a complex variable z, such as, for example, $p(z) = z^{-1} + 1 + 2z$. Here, the *adjoint* $p^*(z)$ stands simply for

$$p^*(z) = p(z^{-1}) \tag{6}$$

and the operation results in flipping all roots across the unit circle. If the two-sided polynomial is *symmetric*, then $p^*(z) = p(z)$ and its roots are symmetrically placed about the unit circle.

In its discrete-time version, the spectral factorization problem is stated as follows: Given a symmetric two-sided polynomial $b(z)$ that meets the conditions of symmetry

$$b^*(z) = b(z) \tag{7}$$

and positiveness (here on the unit circle)

$$b(e^{i\omega}) > 0 \quad \text{real } \omega, \ -\pi < \omega \leq \pi, \tag{8}$$

find a real polynomial $x(z)$ in nonnegative powers of z to satisfy

$$x(z)x^*(z) = b(z) \tag{9}$$

and

$$x(z) \neq 0, \quad |z| \geq 1. \tag{10}$$

By (10), the spectral factor is a stable polynomial in the discrete-time (Schur) sense.

Example 2 For

$$b(z) = 2z^{-2} + 6z^{-1} + 9 + 6z + 2z^2$$

$$= 2z^{-2}(z + 0.5 + 0.5j)(z + 0.5 - 0.5j)$$

$$(z + 1 + j)(z + 1 - j)$$

$$= 4(z + 0.5 + 0.5j)(z + 0.5 - 0.5j)$$

$$\times \ (z^{-1} + 0.5 + 0.5j)$$

$$(z^{-1} + 0.5 - 0.5j)$$

(9) yields

$$x(z) = 1 + 2z + 2z^2 = 2(z + 0.5 + 0.5j)$$

$$(z + 0.5 - 0.5j)$$

as the desired spectral factor.

When the right-hand side $b(z)$ possesses some roots on the unit circle, this problem turns out to be unsolvable as (8) fails. If necessary, a less restrictive formulation can then be applied replacing (8) by $b(e^{i\omega}) \geq 0$ and with $x(z) \neq 0$ only for $|z| > 1$ instead of (10). Clearly, the unit-circle roots of $b(z)$ must then appear both in $x(z)$ and $x^*(z)$.

When formulated as above, the spectral factorization problem is always solvable and its solution is unique up to the change of sign (if x is a solution, so is $-x$ and no other solutions exist).

Polynomial Matrix Spectral Factorization

Matrix version of the problem has been encountered since 1960s. In the world of continuous-time problems, real polynomial matrices in a single complex variable s are used. For such a real polynomial matrix $P(s)$, its *adjoint* $P^*(s)$ is defined as

$$P^*(s) = P^T(-s). \tag{11}$$

A polynomial matrix $P(s)$ is *symmetric* or, more precisely, *para-Hermitian*, if $P^*(s) = P(s)$. Needless to say, only square polynomial matrices can be symmetric.

The matrix spectral factorization problem is defined as follows: Given a symmetric polynomial matrix $B(s)$,

$$B^*(s) = B(s), \tag{12}$$

that is also positive definite on the imaginary axis

$$B(i\omega) > 0 \quad \text{for all real } \omega, \tag{13}$$

find a square real polynomial matrix $X(s)$, which satisfies

$$X(s)X^*(s) = B(s) \tag{14}$$

and has no zeros in the closed right half plain Re $s \geq 0$. Such an $X(s)$ is then called *a left spectral factor* of $B(s)$. A *right spectral factor* $Y(s)$ is defined similarly by replacing (14) with

$$Y^*(s)Y(s) = B(s). \tag{15}$$

Example 3 For a symmetric matrix

$$B(s) = \begin{bmatrix} 2 - s^2 & -2 - s \\ -2 + s & 4 - s^2 \end{bmatrix},$$

we have

$$X(s) = \begin{bmatrix} 1.4 + s & -0.2 \\ -1.2 & 1.6 + s \end{bmatrix}$$

as a left spectral factor and

$$Y(s) = \begin{bmatrix} 1 + s & 0 \\ -1 & 2 + s \end{bmatrix}$$

as a right one.

As in the scalar case, less restrictive definitions are sometimes used where the given right-hand side matrix $B(s)$ is only nonnegative definite on the imaginary axis and so the spectral factor is free of zeros in the open right half plain Re $s > 0$ only.

In the kingdom of discrete-time, *two-sided real polynomial* matrices $P(z)$ are used having in general entries with both positive and negative powers of the complex variable z. For such a matrix, its *adjoint* $P^*(z)$ is defined by

$$P^*(z) = P^T(z^{-1}). \tag{16}$$

Clearly, if $P(z)$ has only nonnegative powers of z, then $P^*(z)$ has only nonpositive powers of z and vice versa. A square two-sided polynomial matrix $P(z)$ is (*para-Hermitian*) *symmetric* if $P^*(z) = P(z)$.

Here is the discrete-time version of matrix spectral factorization problem. Given a two-sided polynomial matrix $B(z)$ that is symmetric

$$B^*(z) = B(z) \tag{17}$$

and positively definite on the unit circle

$$B(e^{i\omega}) > 0 \quad \text{real } \omega, \ -\pi < \omega \leq \pi, \tag{18}$$

find a real polynomial matrix $X(z)$ in nonnegative powers of z such that

$$X(z)X^*(z) = B(z) \tag{19}$$

and has no zeros on and outside of the unit circle. Such an $X(z)$ is then called *a left spectral factor* of $B(z)$. A *right* (The right and the left spectral factor are sometimes called the *factor* and the *cofactor*, respectively, but the terminology is not set at all.) *spectral factor* $Y(z)$ is defined similarly by replacing (19) with

$$Y^*(z)Y(z) = B(z) \tag{20}$$

Example 4 A symmetric two-sided polynomial matrix

$$B(z) = \begin{bmatrix} -2z^{-1} + 5 - 2z & 2z^{-1} - 1 \\ -1 + 2z & 2z^{-1} + 6 + 2z \end{bmatrix}$$

has a left spectral factor

$$X(z) \cong \begin{bmatrix} -1.1 + 1.9z & 0.55 \\ -0.8z & 0.95 + 2.1z \end{bmatrix}$$

and a right spectral factor

$$Y(z) = \begin{bmatrix} 2z - 1 & 1 \\ 0 & 1 + 2z \end{bmatrix}.$$

As before, less restrictive formulations are sometimes encountered where the given symmetric $B(z)$ is only nonnegatively definite on the unit circle and so the spectral factor must have no zeros only outside of the unit circle.

When formulated as above, the matrix spectral factorization problem is always solvable. The spectral factors are unique up to an orthogonal matrix multiple. That is, if X and X' are two left spectral factors of B, then

$$X' = UX \tag{21}$$

where U is a constant orthogonal matrix $UU^T = I$, while if Y and Y' are two right spectral factors of B, then

$$Y' = YV \qquad (22)$$

where V is a constant orthogonal matrix $V^T V = I$.

J-Spectral Factorization

In robust control, game theory and several other fields, the symmetric right-hand side in the matrix spectral factorization may have a general signature. With such a right-hand side, standard (positive or nonnegative definite) factorization becomes impossible. Here, a similar yet different J-spectral factorization takes its role.

In the context of continuous-time problems, the *J-spectral factorization problem* is formulated as follows. Given a symmetric polynomial matrix $B(s)$,

$$B^*(s) = B(s), \qquad (23)$$

find a square real polynomial matrix $X(s)$, which satisfies

$$X(s)JX^*(s) = B(s), \qquad (24)$$

where $X(s)$ has no zeros in the open right half plain Re s > 0 and J is a *signature matrix* of the form

$$J = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & -I_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (25)$$

with I_1 and I_2 unit matrices of not necessarily the same dimensions. The bottom right block of zeros is often missing, yet it is considered here for generality. Such an $X(s)$ is called *a left J-spectral factor* of $B(s)$. A *right J-spectral factor* is defined by

$$Y^*(s)J Y(s) = B(s) \qquad (26)$$

instead of (24). For discrete-time problems, the J-spectral factorization is defined analogously.

The *J-spectral factorization* problem is quite general having standard (either positive or nonnegative) spectral factorization as a particular case. No necessary and sufficient existence conditions appear to be known for J-spectral factorization. A sufficient condition by Jakubovič (1970) states that the problem is solvable if the multiplicity of the zeros on the imaginary axis of each of the invariant polynomials of the right-hand side matrix is even. In particular, this condition is satisfied whenever det $B(s)$ has no zeros on the imaginary axis. In turn, the condition is violated if any of the invariant factors is not factorable by itself. An example of a nonfactorizable polynomial is $1 + s^2$.

The J-spectral factors are unique up to a J-orthogonal matrix multiple. That is, if X and X' are two left J-spectral factors of B, then

$$X' = UX, \qquad (27)$$

where U is a J-orthogonal matrix $UJU^T = J$, while if Y and Y' are two right J-spectral factors of B, then

$$Y' = YV, \qquad (28)$$

where V is a J-orthogonal matrix $V^T J V = J$.

Example 5 For

$$B(s) = \begin{bmatrix} 0 & 1-s \\ 1+s & 2-s^2 \end{bmatrix}$$

the signature matrix reads

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

and the right J-spectral factor is

$$Y(s) = \begin{bmatrix} 1+s & \dfrac{3-s^2}{2} \\[2mm] 1+s & \dfrac{1-s^2}{2} \end{bmatrix}$$

Nonsymmetric Spectral Factorization

Spectral factorization can also be non-symmetric. For a scalar polynomial p (either in s or in z), this means to factor it directly as

$$p = p^+ p^- \qquad (29)$$

where p^+ is a stable factor of p (having all its roots either in the open left half plane or inside of the unit disc, depending on the variable type) while p^- is the "remaining" that is unstable factor. Eventual roots of p at the stability boundary either associate to p^+ or to p^-, depending on the application problem at hand.

For a matrix polynomial P, the non-symmetric factorization is naturally twofold: Either

$$P = P^+ P^- \qquad (30)$$

or

$$P = P^- P^+. \qquad (31)$$

For scalar polynomials, symmetric and non-symmetric spectral factors are closely related. Given p and having computed a symmetric factor x for pp^* as in (4) or (9) to get

$$x^* x = p^* p \qquad (32)$$

Then

$$p^+ = \gcd(p, x) \quad \text{and} \quad p^- = \gcd(p, x^*) \quad (33)$$

where *gcd* stands for a greatest common divisor. In reverse,

$$x = p^+ (p^-)^* \text{ and } x^* = p^- (p^+)^*. \qquad (34)$$

Unfortunately, no such relations exist for the matrix case.

Example 6 For example,

$$p(s) = 1 - s^2$$

factorizes into

$$p^+(s) = 1 + s, \quad p^-(s) = 1 - s$$

while for

$$P(s) = \begin{bmatrix} 1 + s & 0 \\ 1 + s^2 & 1 - s \end{bmatrix}$$

we have

$$P^-(s) = \begin{bmatrix} 1 & 1 \\ s & 1 \end{bmatrix}, P^+(s) = \begin{bmatrix} s & -1 \\ 1 & 1 \end{bmatrix}.$$

Algorithms and Software

Spectral factorization is a crucial step in the solution of various control, estimation, filtration, and other problems. It is no wonder that a variety of methods has been developed over the years for the computation of spectral factors. The most popular ones are briefly mentioned here. For details on particular algorithms, the reader is referred to the papers recommended for further reading.

Factor Extraction Method
If all roots of the right-hand side polynomial are known, the factorization becomes trivial. Just write the right-hand side as a product of first and second order factors and then collect the stable ones to create the stable factor. If the roots are not known, one can first enumerate them and then proceed as above. Somewhat surprisingly, a similar procedure can be used for the matrix case. To every zero, a proper matrix factor must be extracted. For further details, see Callier (1985) or Henrion and Sebek (2000).

Bauer's Algorithm
This procedure is an iterative scheme with linear rate of convergence. It relies on equivalence between the polynomial spectral factorization and the Cholesky factorization of a related infinite-dimensional Toeplitz matrix. For further details, see Youla and Kazanjian (1978).

Newton-Raphson Iterations
An iterative algorithm with quadratic convergence rate based on consecutive solutions of symmetric linear polynomial Diophantine equations. It is inspired by the classical Newton's method for finding a root of a function. To learn more, read Davis (1963), Ježek and Kučera (1985), Vostrý (1975).

Factorization via Riccati Equation
In state-space solution of various problems, an algebraic Riccati equation plays the role of spectral factorization. It is therefore not surprising that the spectral factor itself can directly be calculated by solution of a Riccati equation. For further info, see e.g. Šebek (1992).

FFT Algorithm

This is the most efficient and accurate procedure for factorization of scalar polynomials with very high degrees (in orders of hundreds or thousands). Such polynomials appear in some special problems of signal processing in advanced audio applications involving inversions of dynamics of loudspeakers or room acoustics. The algorithm is based on the fact that logarithm of a product (such as the spectral factorization equation) turns into a sum of logarithms of particular entries. For details, see Hromčík and Šebek (2007)

All the procedures above are either directly programmed or can be easily composed from the functions of *Polynomial Toolbox for Matlab*, which is a third-party Matlab toolbox for polynomials, polynomial matrices and their applications in systems, signals, and control. For more details on the toolbox, visit *www.polyx.com*.

Consequences and Comments

Polynomial and polynomial matrix spectral factorization is an important step when frequency domain (polynomial) methods are used for optimal and robust control, filtering, estimation, or prediction. Numerous particular examples can be found throughout this Encyclopedia as well as in the textbooks and papers recommended for further reading below.

Spectral factorization of rational functions and matrices is an equally important topic but it is omitted here due to lack of space. Inquiring readers are referred to the papers Oara and Varga (2000) and Zhong (2005).

Cross-References

Recommended Reading

Nice tutorial books on polynomials and polynomial matrices in control theory and design are Kučera (1979), Callier and Desoer (1982), and Kailath (1980)

The concept of spectral factorization was introduced by Wiener (1949), for further information see later original papers Wilson (1972) or Kwakernaak and Šebek (1994) as well as survey papers Kwakernaak (1991), Sayed and Kailath (2001) or Kučera (2007).

Nice applications of spectral factorization in control problems can be found e.g. in Green et al. (1990), Henrion et al. (2003) or Zhou and Doyle (1998). For its use of in other engineering problems see e.g. Sternad and Ahlén (1993).

Bibliography

Callier FM (1985) On polynomial matrix spectral factorization by symmetric extraction. IEEE Trans Autom Control 30:453–464

Callier FM, Desoer CA (1982) Multivariable feedback systems. Springer, New York

Davis MC (1963) Factorising the spectral matrix. IEEE Trans Autom Control 8:296

Green M, Glover K, Limebeer DJN, Doyle J (1990) A *J*-spectral factorization approach to *H*-infinity control. SIAM J Control Opt 28:1350–1371

Henrion D, Sebek M (2000) An algorithm for polynomial matrix factor extraction. Int J Control 73(8):686–695

Henrion D, Šebek M, Kučera V (2003) Positive polynomials and robust stabilization with fixed-order controllers. IEEE Trans Autom Control 48:1178–1186

Hromčík M, Šebek M (2007) Numerical algorithms for polynomial Plus/Minus factorization. Int J Robust Nonlinear Control 17(8):786–802

Jakubovič VA (1970) Factorization of symmetric matrix polynomials. Dokl. Akad. Nauk SSSR, 194(3):532-535

Ježek J, Kučera V (1985) Efficient algorithm for matrix spectral factorization. Automatica 29: 663–669

Kailath T (1980) Linear systems. Prentice-Hall, Englewood Cliffs

Kučera V (1979) Discrete linear control: the polynomial equation approach. Wiley, Chichester

Kučera V (2007) Polynomial control: past, present, and future. Int J Robust Nonlinear Control 17:682–705

Kwakernaak H (1991) The polynomial approach to a H-optimal regulation. In: Mosca E, Pandolfi L (eds) H-infinity control theory. Lecture Notes in Maths, vol 1496. Springer, Berlin

Kwakernaak H, Šebek M (1994) Polynomial J-spectral factorization. IEEE Trans Autom Control 39:315–328

Oara C, Varga A (2000) Computation of general inner-outer and spectral factorizations. IEEE Trans Autom Control 45:2307–2325

Sayed AH, Kailath T (2001) A survey of spectral factorization methods. Numer Linear Algebra Appl 8(6–7):467–496

Sternad M, Ahlén A (1993) Robust filtering and feed-forward control based on probabilistic descriptions of model errors. Automatica 29(3):661–679

Šebek M (1992) J-spectral factorization via Riccati equation. In: Proceedings of the 31st IEEE CDC, Tuscon, pp 3600–3603

Vostrý Z (1975). New algorithm for polynomial spectral factorization with quadratic convergence. Kybernetika 11:415, 248

Wiener N (1949) Extrapolation, interpolation and smoothing of stationary time series. Wiley, New York

Wilson GT (1972) The factorization of matricial spectral densities. SIAM J Appl Math 23:420

Youla DC, Kazanjian NN (1978) Bauer-type factorization of positive matrices and the theory of matrix polynomials orthogonal on the unit circle. IEEE Trans Circuits Syst 25:57

Zhong QC (2005) J-spectral factorization of regular para-Hermitian transfer matrices. Automatica 41: 1289–1293

Zhou K, Doyle JC (1998) Essentials of robust control. Prentice-Hall, Upper Saddle River

Stability and Performance of Complex Systems Affected by Parametric Uncertainty

Boris Polyak and Pavel Shcherbakov
Institute of Control Science, Moscow, Russia

Abstract

Uncertainty is an inherent feature of all real-life complex systems. It can be described in different forms; we focus on the parametric description. The simplest results on stability of linear systems under parametric uncertainty are the Kharitonov theorem, edge theorem, and graphical tests. More advanced results include sufficient conditions for robust stability with matrix uncertainty, LMI tools, and randomized methods. Similar approaches are used for robust control synthesis, where performance issues are crucial.

Keywords

Edge theorem; Kharitonov theorem; Linear systems; Matrix; Parametric uncertainty and robustness; Quadratic stability; Randomized methods; Robust and optimal design; Robust stability; Tsypkin–Polyak plot

Introduction

Mathematical models for systems and control are often unsatisfactory due to the incompleteness of the parameter data. For instance, the ideas of off-line optimal control can only be applied to real systems if all the parameters, exogenous perturbations, state equations, etc. are known precisely. Moreover, feedback control also requires a detailed information which is not available in most cases. For example, to drive a car with four-wheel control, the controller should be aware of the total weight, location of the center of gravity, weather conditions, and highway properties as well as many other data which may not be known. In that respect, even such a relatively simple real-life system can be considered a *complex* one; in such circumstances, control under uncertainty is a highly important issue.

The focus in this article is on the *parametric uncertainty*; other types of uncertainty can be treated in more general models of robustness. This topic became particularly popular in the control community in the mid- to late 1980s of the previous century; at large, the results of this activity have been summarized in the monographs (Ackermann 1993; Barmish 1994; Bhattacharyya et al. 1995).

We start with problems of stability of polynomials with uncertain parameters and present the simplest robust stability results for this case together with the most important machinery. Next, we consider stability analysis for the matrix uncertainty; most of the results are just sufficient conditions. We present some useful tools for the analysis, such as the LMI technique and randomized methods. Robust control under parametric uncertainty is the next step; we briefly discuss several problem formulations for this case.

Stability of Linear Systems Subject to Parametric Uncertainty

Consider the closed-loop linear, time invariant continuous time state space system

$$\dot{x} = Ax, \qquad x(0) = x_0, \qquad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, x_0 is an arbitrary finite initial condition, and $A \in \mathbb{R}^{n \times n}$ is the state matrix. The system is stable (i.e., no matter what x_0 is, the solutions tend to zero as $t \to \infty$) if and only if all eigenvalues λ_i of the matrix A have negative real parts:

$$\mathrm{Re}\,\lambda_i < 0, \qquad i = 1, \dots, n, \qquad (2)$$

in which case, A is said to be a *Hurwitz* matrix. If it is known precisely, checking condition (2) is immediate. For instance, one might compute the characteristic polynomial

$$p(s) = \det(sI - A) = a_0 + a_1 s + \cdots +$$
$$a_{n-1}s^{n-1} + s^n \qquad (3)$$

of A (here, I is the identity matrix) and use any of the stability tests (e.g., the Routh algorithm, Routh–Hurwitz test, and graphical tests such as the Mikhailov plot or Hermite–Biehler theorem), see Gantmacher (2000). Alternatively, the eigenvalues can be directly computed using the currently available software, such as MATLAB.

However, things get complicated if the knowledge of the matrix A is incomplete; for instance,

it can depend on the (real) parameters $q = (q_1, \dots, q_m)$ which take arbitrary values within the given intervals:

$$A = A(q), \qquad \underline{q}_i \le q_i \le \overline{q}_i, \quad i = 1, \dots, m. \qquad (4)$$

In that case, we arrive at the *robust stability problem*; i.e., the goal is to check if condition (2) holds for *all matrices* in the family (4).

The two main components of any robust stability setup are the *feasible set* $\mathcal{Q} \subset \mathbb{R}^\ell$, in which the uncertain parameters are allowed to take their values (usually a ball in some norm; e.g., the box as in (4)), and the *uncertainty structure*, which defines the functional dependence of the coefficients on the uncertain parameters. Of the most interest are the affine and multiaffine dependence; typically, more general situations are hard to handle.

Simple Solutions

In some cases, the robust stability problem admits a simple solution. Perhaps the most striking example is the so-called Kharitonov theorem (Kharitonov 1978); also see Barmish (1994), where this seminal result is referred to as a *spark* because of its transparency and elegance.

Namely, consider the *interval polynomial family*

$$\mathcal{P} = \{p(s) = q_0 + q_1 s + \cdots + q_n s^n,$$
$$\underline{q}_i \le q_i \le \overline{q}_i, \quad i = 0, \dots, n\}, \quad (5)$$

where the coefficients q_i are allowed to take values in the respective intervals *independently of each other* and distinguish the following four elements in this family:

$$p_1(s) = \underline{a}_0 + \underline{q}_1 s + \overline{q}_2 s^2 + \overline{q}_3 s^3 + \dots$$
$$p_2(s) = \underline{q}_0 + \overline{q}_1 s + \overline{q}_2 s^2 + \underline{q}_3 s^3 + \dots$$
$$p_3(s) = \overline{q}_0 + \overline{q}_1 s + \underline{q}_2 s^2 + \underline{q}_3 s^3 + \dots$$
$$p_4(s) = \overline{q}_0 + \underline{q}_1 s + \underline{q}_2 s^2 + \overline{q}_3 s^3 + \dots$$

By the Kharitonov theorem, the interval family (5) is robustly stable (i.e., all polynomials

in (5) are Hurwitz having all roots with negative real parts) if and only if the *four Kharitonov polynomials*, p_1, p_2, p_3, and p_4, are Hurwitz.

A simple and transparent proof of this result can be obtained using the *value set concept* (Zadeh and Desoer 1963) and the *zero exclusion* principle (Frazer and Duncan 1929), the two general tools which are in the basis of many results in the area of robust stability. We illustrate these concepts via robust stability of polynomials.

Given the uncertain polynomial family

$$\mathcal{P}(s, Q) = \{p(s, q), \quad q \in \mathcal{Q}\},$$

the set

$$\mathcal{V}(\omega) = \{p(j\omega, q): \ \omega \geq 0, \ q \in \mathcal{Q}\}$$

is referred to as the *value set*, which is, by definition, the set on the complex plane obtained by fixing the argument s to be $j\omega$ for a certain value of ω and letting the uncertain parameter vector q sweep the feasible domain.

The zero exclusion principle states that, under certain regularity requirements, the uncertain polynomial family is robustly stable if and only if it contains a stable element and the following condition holds:

$$0 \notin \mathcal{V}(\omega) \quad \forall \, \omega \geq 0. \tag{6}$$

To use this machinery, one has to be able to compute efficiently the value set and check condition (6). For the interval family (5), the value set can be shown to be the rectangle with coaxial edges and the vertices being the values of the four Kharitonov polynomials; see Fig. 1.

Being an extremely propelling result, the Kharitonov theorem is not free of drawbacks. First of all, it is not capable of determining the maximal lengths of the uncertainty intervals that retain the robust stability. This relates to an important notion of *robust stability margin*; for simplicity, we define this quantity for the case of the interval family (5). Namely, introduce the *nominal polynomial* $p_0(s)$ with coefficients

$$q_i^0 = (\overline{q}_i + \underline{q}_i)/2,$$
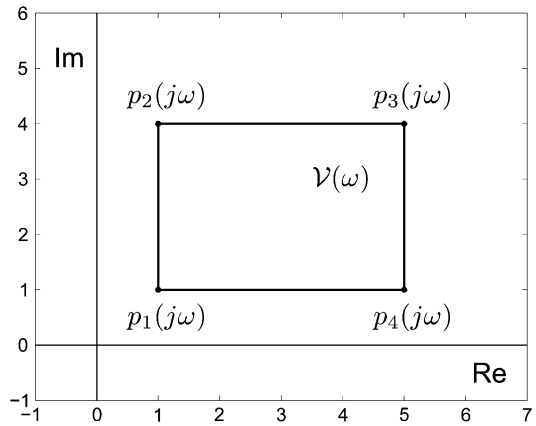
Stability and Performance of Complex Systems Affected by Parametric Uncertainty, Fig. 1 The Kharitonov rectangular value set

and the *scaling factors*

$$\alpha_i = (\overline{q}_i - \underline{q}_i)/2$$

for the deviations of the coefficients. Then the robust stability margin $r_{\max}$ is defined as follows:

$$r_{\max} = \sup\{r: \ p(s, q) \ (5) \text{ is stable } \forall \, q_i:$$
$$|q_i - q_i^0| \leq r\alpha_i, \quad i = 1, \ldots, n\}. \tag{7}$$

Anther drawback of the Kharitonov result is its inapplicability to the discrete-time case (Schur stability of polynomials).
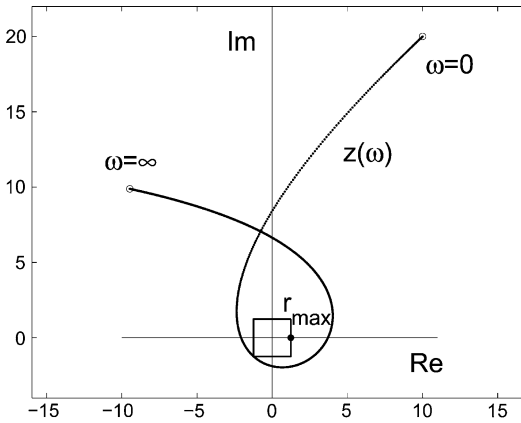
A more flexible graphical test for robust stability uses the so-called Tsypkin–Polyak plot (Tsypkin and Polyak 1991), which is defined as the parametric curve on the complex plane:

$$z(\omega) = x(\omega) + jy(\omega), \ j = \sqrt{-1}; \ 0 \leq \omega < \infty,$$

where

$$x(\omega) = \frac{q_0^0 - q_2^0\omega^2 + \ldots}{\alpha_0 + \alpha_2\omega^2 + \ldots},$$

$$y(\omega) = \frac{q_1^0 - q_3^0\omega^2 + \ldots}{\alpha_1 + \alpha_3\omega^2 + \ldots}. \tag{8}$$

Then, by the Tsypkin–Polyak criterion, the polynomial family (5) is robustly stable if and only if the following conditions hold: (i) $q_0^0 > \alpha_0$,

Stability and Performance of Complex Systems Affected by Parametric Uncertainty, Fig. 2 The Tsypkin–Polyak plot

$q_n^0 > \alpha_n$, and (ii) as ω changes zero to infinity, the curve $z(\omega)$ goes consecutively through n quadrants in the counterclockwise direction and does not intersect the unit square with the vertices $(\pm 1, \pm j)$.

Unlike the Kharitonov theorem, with this test, the robust stability margin of family (5) can be determined as the size of the maximal square inscribed in the curve $z(\omega)$; see Fig. 2. Moreover, with minor modifications, this test applies to *dependent uncertainty structures* where the coefficient vector $q = (q_0, \ldots, q_n)^\top$ is confined to a ball in ℓ_p-norm, not to a *box* as in (5).

On top of that, the Tsypkin–Polyak plot can be built for discrete-time systems which do not admit any counterparts of the Kharitonov theorem.

It is fair to say that interval polynomial families is an idealization, since the coefficients of the characteristic polynomial can hardly be thought of as the physical parameters of the real-world system. As a step towards more realistic formulations, consider the *affine polynomial family* of the form

$$p(s) = p_0(s) + \sum_{i=1}^{m} q_i \, p_i(s), \quad |q_i| \leq 1,$$
$$i = 1, \ldots, m, \quad (9)$$

where p_i are the given polynomials and the q_is are the uncertain parameters (clearly, they can

be scaled to take values in the segment $[-1, 1]$). The famous *edge theorem* (Bartlett et al. 1988) claims that checking the robust stability of such a family is equivalent to checking the *edges* of the uncertainty box, i.e., the points $q \in \mathbb{R}^m$ with all but one components being fixed to ± 1, while the "free" coordinate varies in $[-1, 1]$.

Complex Solutions

Obviously, the affine model (9) covers just a small part of problems with parametric uncertainty. Closed-form solutions cannot be obtained in the general case; however, many important classes of systems can be analyzed efficiently.

Thus, in the engineering practice, *block diagram description* of systems is often more convenient than differential equations of the form (1). The blocks are associated with typical elements such as amplifiers, integrators, lag elements, and oscillators, which are connected in a certain circuit. In this case, transfer functions are the most adequate tool for dealing with such systems. For instance, the transfer function of the lag element is given by

$$W(s) = 1/(Ts + 1),$$

where the scalar T is the *time constant* of the element. In terms of differential equations, this means that the input $u(t)$ of a block and its output $x(t)$ satisfy the equation $T\dot{x} + x = u$.

Assume now we have a set of m cascade connected elements with uncertain time constants

$$\underline{T}_i \leq T_i \leq \overline{T}_i, \quad i = 1, \ldots, m, \quad (10)$$

with known lower and upper bounds. The characteristic polynomial of such a connection embraced by the feedback with *gain k* is known to have the form

$$p(s) = k + (1 + T_1 s) \cdots (1 + T_m s). \quad (11)$$

Hence, the robust stability problem reduces to checking if all polynomials (11) with constraints (10) are Hurwitz. Note that the coefficients of such a polynomial depend *multilinearly* on the uncertain parameters T_i

(cf. linear dependence in (9)), making the problem much more complicated.

The solution of the problem above was obtained in Kiselev et al. (1997) for many important special cases; the closely related problem of finding the "critical gain" (the maximal value of k retaining the robust stability) was also addressed.

Using the similar technique, closed-form solutions can be obtained for a number of similar problems such as robust sector stability, robust stability of distributed systems, robust D-decomposition, to name just a few.

Difficult Problems: Possible Approaches

In spite of the apparent progress obtained in the area of parametric robustness, the list of unsolved problems is still quite large. Moreover, some of the formulations were shown to be NP-hard, making it hard to believe that any efficient solution methods will ever be found.

One of such fundamental problems is robust stability of the *interval matrix*. Specifically, assume that the entries a_{ij} of the matrix A in (1) are interval numbers

$$\underline{a}_{ij} \le a_{ij} \le \overline{a}_{ij}, \qquad i, j = 1, \ldots, n;$$

the problem is to check if the interval matrix is robustly stable, i.e., if the eigenvalues of all matrices in this family have negative real parts. Numerous attempts to prove a Kharitonov-like theorem for matrices have failed, and the results by Nemirovskii (1994) on NP-hardness showed that these generalizations are not possible. It was also shown that the edge theorem for matrix families is not valid. The other NP-hard problems in robustness include the analysis of systems with interval delays, parallel connection of uncertain blocks, problem (11)–(10) with nested segments $[\underline{T}_i, \overline{T}_i]$, and others.

However, a change in the statement of the problem often allows for simple and elegant solutions. We mention three fruitful reformulations.

In the first approach, the uncertain parameters are assumed to have random rather than deterministic nature; for instance, they are assumed to be uniformly distributed over the respective intervals of uncertainty. We next specify an acceptable

tolerance ε, say $\varepsilon = 0.01$, and check if the resulting random family of polynomials is stable with probability no less than $(1 - \varepsilon)$; see Tempo et al. (2013) for a comprehensive exposition of such a *randomized approach to robustness*.

In many of the NP-hard robustness problems, such a reformulation often leads to exact or approximate solutions. Moreover, the randomized approach has several attractive properties even in the situations where the deterministic solution is available. Indeed, the deterministic statements of robustness problems are minimax; hence, the answer is dictated by the "worst" element in the family, whereas these critical values of the uncertain parameters are rather unlikely to occur. Therefore, by neglecting a small risk of violation of the stability, the admissible domains of variation of the parameters may be considerably extended. This effect is known as the *probabilistic enhancement of robustness margins*; it is particularly tangible for the large number of the parameters. Another attractive property of the randomized approach is its low computational complexity which only slowly grows with increase of the number of uncertain parameters.

To illustrate, let us turn back to problem (11)–(10) and use the value set approach. In the considered problem, this set can be efficiently built.

Assume now that the parameters T_i are independent random variables uniformly distributed over the respective segments (10) and consider the random variable

$$\eta = \eta(\omega) = \log(p(j\omega) - k) = \sum_{i=1}^{m} \log(1 + j\omega T_i). \tag{12}$$

The right-hand side of the last relation is the sum of independent complex-valued random variables; for m large, its behavior obeys the central limit theorem, so that the probability that η belongs to the respective *confident ellipse* $\mathcal{E} = \mathcal{E}(\omega)$ is close to unity. In other words, we have

$$p(j\omega) \approx k + e^{\mathcal{E}} \doteq \mathcal{G}(\omega),$$

and the set $\mathcal{G}(\omega)$ is referred to as a *probabilistic predictor* of the value set $\mathcal{V}(\omega)$; it is the shifted

set of points of the form $e^z, z \in \mathcal{E} \subset \mathbb{C}$. The predictor $\mathcal{G}(\omega)$ constitutes a small portion of the deterministic value set $\mathcal{V}(\omega)$, yielding the probabilistic enhancement of the robustness margin.

Note also that the computation of $\mathcal{E}$ and $e^{\mathcal{E}}$ is nearly trivial and, in contrast to the construction of the true value set $\mathcal{V}$, the complexity does not grow with increase of m.

The second approach to solving "hard" problems in robust stability relates to the notion of *superstability* (Polyak and Shcherbakov 2002). The matrix A of system (1) (and the system itself) is said to be superstable, if its entries a_{ij}, $i, j = 1, \ldots, n$, satisfy the relations

$$a_{ii} < 0, \qquad \min_i(-a_{ii} - \sum_{j \neq i} |a_{ij}|) = \sigma > 0.$$

The following estimate holds for the solutions of the superstable system (1):

$$\|x(t)\|_\infty \leq \|x(0)\|_\infty e^{-\sigma t},$$

i.e., it is stable, and the (nonsmooth) function $\|x\|_\infty$ is a Lyapunov function for the system. Since the condition of superstability is formulated in terms of linear inequalities on the entries of A, checking robust superstability of affine (and in particular, interval) matrix families is immediate. Similar situation holds for so-called positive systems.

The third approach to robustness analysis relates to *quadratic stability* (Leitmann 1979; Boyd et al. 1994). Namely, a family of systems is said to be *robustly quadratically stable* if it possesses a common quadratic Lyapunov function $V(x) = x^\top P x$ with positive definite matrix P. In other words, an uncertain family of matrices $A(q), q \in \mathcal{Q}$ has to satisfy the following set of the matrix Lyapunov-type inequalities:

$$A(q)P + PA(q)^\top \prec 0, \quad q \in \mathcal{Q}, \quad P \succ 0, \tag{13}$$

where the symbols $\prec, \succ$ stand for the sign-definiteness of a matrix.

The inequality above is referred to as a *linear matrix inequality* (LMI), (Boyd et al. 1994); there exist both efficient numerical methods for solving

such inequalities (*interior point methods*) and various software, e.g., MATLAB. This approach can be directly applied at least in the following two cases: (i) the set $\mathcal{Q}$ contains a finite number of points and (ii) $\mathcal{Q}$ is a polyhedron and the dependence $A(q)$ is affine. In the general setup or in the high-dimensional problems, randomized methods can be employed.

Finding the *quadratic robust stability margin* (by analogy with the stability margin, this is the maximum span of the feasible set $\mathcal{Q}$ that allows for the existence of the common Lyapunov function) in this problem is also possible; it reduces to the minimization of a linear function over the solutions of a similar LMI.

Note that the approaches based on superstability and quadratic stability provide only sufficient conditions for robustness.

Robust Control

So far, of our primary interest was in assessing the robust stability of a closed-loop system with synthesized linear feedback. A more important problem is to *design* a controller that makes the closed-loop system robustly stable and guarantees certain *robust performance* of the system.

Robust Stabilization
Let the linear system

$$\dot{x} = A(q)x + Bu$$

depend on the vector $q \in \mathcal{Q}$ of uncertain parameters. In the simplest form, the problem of *robust stabilization* consists in finding the linear static state feedback

$$u = Kx$$

that guarantees the robust stability of the closed-loop system. Alternatively, static or dynamic *output* robustly stabilizing controllers can be considered in the situations where only the linear output $y = Cx$ of the system is available, but not the complete state vector x.

If the number of controller parameters to be tuned is small (which is the case for PI or PID controllers), then the design can be accomplished using the *D-decomposition technique*.

In the general formulation, the problem of robust design is complicated; it can, however, be addressed with the use of randomized methods (Tempo et al. 2013). Other plausible approaches include superstability and quadratic stability; respectively, the problem reduces to solving linear programs or linear matrix inequalities in the coefficients of the controller.

Robust Performance

Needless to say, the robust stabilization problem is not the only one in the area of optimal control. As a rule, a certain cost function is always involved (say, integral quadratic), and its desired value should be guaranteed for all admissible values of the uncertain parameters. Moreover, robust stability is a necessary condition for such a guaranteed estimate to exist. This sort of problems can often be cast in the form of LMIs which must be satisfied for all admissible values of the parameters. Such robust LMIs can be solved either directly or using various randomized techniques presented in Tempo et al. (2013).

Conclusions

In spite of the considerable progress attained in the parametric robustness of complex systems, this topic is still a vivid and active research area. To date, randomization, superstability, and quadratic stability present the most efficient and diverse tools for the analysis and design of systems affected by parametric uncertainty.

Cross-References

▶ H-Infinity Control
▶ LMI Approach to Robust Control
▶ Optimization Based Robust Control
▶ Randomized Methods for Control of Uncertain Systems

Bibliography

Ackermann J (1993) Robust control: systems with uncertain physical parameters. Springer, London

Barmish BR (1994) New tools in robustness of linear systems. Macmillan, New York

Bartlett AC, Hollot CV, Lin H (1988) Root locations of an entire polytope of polynomials: it suffices to check the edges. Math Control Sig Syst 1(1):61–71

Bhattacharyya SP, Chapellat H, Keel LH (1995) Robust control: the parametric approach. Prentice Hall, Upper Saddle River

Boyd S, El Ghaoui L, Feron E, Balakrishnan V (1994) Linear matrix inequalities in system and control theory. SIAM, Philadelphia

Frazer RA, Duncan WJ (1929) On the criteria for the stability of small motions. Proc R Soc Lond A 124(795):642–654

Gantmacher FR (2000) The theory of matrices. AMS, Providence

Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equations. Differentsial'nye Uravneniya 14:2086–2088

Kiselev ON, Le Hung Lan, Polyak BT (1997) Frequency responses under parametric uncertainty. Autom Remote Control 58(Pt. 2, 4):645–661

Leitmann G (1979) Guaranteed asymptotic stability for some linear systems with bounded uncertainties. J Dyn Syst Measure Control 101(3):212–216

Nemirovskii AS (1994) Several NP-hard problems arising in robust stability analysis. Math Control Sig Syst 6(1):99–105

Polyak BT, Shcherbakov PS (2002) Superstable linear control systems. I. Analysis; II. Design. Autom Remote Control 63(8):1239–1254; 63(11):1745–1763

Tempo R, Calafiore G, Dabbene F (2013) Randomized algorithms for analysis and control of uncertain systems, with applications, 2nd edn. Springer, London

Tsypkin YZ, Polyak BT (1991) Frequency domain criteria for l^p-robust stability of continuous systems. IEEE Trans Autom Control 36(12):1464–1469

Zadeh LA, Desoer CA (1963) Linear system theory – a state space approach. McGraw-Hill, New York

S

Stability Theory for Hybrid Dynamical Systems

Andrew R. Teel
Electrical and Computer Engineering
Department, University of California, Santa
Barbara, CA, USA

Abstract

This entry provides a short introduction to modeling of hybrid dynamical systems and then focuses on stability theory for these systems. It provides

definitions of asymptotic stability, basin of attraction, and uniform asymptotic stability for a compact set. It points out mild assumptions under which different characterizations of asymptotic stability are equivalent, as well as when an asymptotically stable compact set exists. It also summarizes necessary and sufficient conditions for asymptotic stability in terms of Lyapunov functions.

Keywords

Asymptotic stability; Basin of attraction; Hybrid system; Lyapunov function

Introduction

A hybrid dynamical system combines continuous change and instantaneous change. Instantaneous change is the only type of change available for variables like counters, switches, and logic variables. Instantaneous change may also be a good approximation of what occurs to velocities in mechanical systems at the time of an impact with a wall, floor, or some other rigid body. At other times, velocities evolve continuously. Continuous change is also natural for position variables, continuous timers, and voltages and currents. For mathematical convenience, it is typical in the analysis of hybrid dynamical systems to embed all of these variables into a Euclidean space, with the understanding that many points in the state space will never be reached. For example, a logic variable that naturally takes values in the set {off, on} is typically embedded in the real number line where its two distinct values are associated with two distinct numbers, the only numbers that this variable will visit during its evolution.

A finite-dimensional dynamical system that exhibits continuous change exclusively is typically modeled by an ordinary differential equation, or sometimes a more flexible differential inclusion. A system that exhibits purely instantaneous change is typically modeled by a difference equation or inclusion. Consequently, a hybrid dynamical system combines a differential equation or inclusion with a difference equation

or inclusion. A big part of the modeling effort for hybrid systems is directed at determining which type of evolution should be allowed at each point in the state space. To this end, subsets of the state space are specified where each type of behavior is allowed, like in the description of the heating system given above.

Though the behavior of a hybrid dynamical system can be quite complex and nonconventional, it is still reasonable to ask the same stability questions for them that might be asked about classical differential or difference equations. Moreover, the same stability analysis tools that are used for classical systems are also quite useful for hybrid dynamical systems. The emphasis of this entry is on basic stability theory for hybrid dynamical systems, focusing on definitions and tools that also apply to classical systems.

Mathematical Modeling

System Data
A hybrid dynamical system with state x belonging to a Euclidean space $\mathbb{R}^n$ combines a differential equation or inclusion, written formally as $\dot{x} = f(x)$ or $\dot{x} \in F(x)$, with a difference equation or inclusion $x^+ = g(x)$ or $x^+ \in G(x)$, where $\dot{x}$ indicates the time derivative and x^+ indicates the value after an instantaneous change. The mapping f or F is called the *flow map*, while the mapping g or G is called the *jump map*. A complete model also specifies where in the state space continuous evolution is allowed and where instantaneous change is allowed. The set where continuous evolution is allowed is called the *flow set* and is denoted C, whereas the set where instantaneous change is allowed is called the *jump set* and is denoted D. The overall model, using inclusions for generality, is written formally as

$$x \in C \qquad \dot{x} \in F(x) \tag{1a}$$

$$x \in D \qquad x^+ \in G(x). \tag{1b}$$

Solutions
It is natural for solutions of (1) to be functions of two different types of time: a variable t that keeps track of the amount of ordinary time that has

elapsed and a variable j that counts the number of jumps. There is a special structure to the types of domains that are allowed. A *compact hybrid time domain* is a set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$, that is, a subset of the product of the nonnegative real numbers and the nonnegative integers, of the form

$$E = \bigcup_{i=0}^{J} ([t_i, t_{i+1}] \times \{i\})$$

for some $J \in \mathbb{Z}_{\geq 0}$ and some sequence of non-decreasing times $0 = t_0 \leq t_1 \leq \cdots \leq t_{J+1}$. It is possible for several of these times to be the same, which would correspond to more than one jump at the given time. A *hybrid time domain* is a set $E \subset \mathbb{R}_{\geq 0} \times \mathbb{Z}_{\geq 0}$ such that for each $(T, J) \in E$, the set $E \cap ([0, T] \times \{0, \ldots, J\})$ is a compact hybrid time domain. In contrast to a compact hybrid time domain, a hybrid time domain may have an infinite number of intervals, or it may have a finite number of intervals with the last one being unbounded or of the form $[t_J, t_{J+1})$; that is, it may be open on the right. A *hybrid arc* is a function x, defined on a hybrid time domain, such that $t \mapsto x(t, j)$ is locally absolutely continuous for each j; in particular, $t \mapsto x(t, j)$ is differentiable for almost every t where it is defined, and this mapping is the integral of its derivative. The notation "dom x" denotes the domain of x. Finally, a hybrid arc is a *solution* of (1) if the following two properties are satisfied:

1. For $\varepsilon > 0$, $(s, j), (s + \varepsilon, j) \in \text{dom } x$ implies that $x(t, j) \in C$ and $\dot{x}(t, j) \in F(x(t, j))$ for almost all $t \in [s, s + \varepsilon]$.
2. $(t, j), (t, j + 1) \in \text{dom } x$ implies that $x(t, j) \in D$ and $x(t, j + 1) \in G(x(t, j))$.

For a hybrid system with no flow dynamics, each solution has a time domain of the form $\{0\} \times \{0, \ldots, J\}$ for some $J \in \mathbb{Z}_{\geq 0}$ or $\{0\} \times \mathbb{Z}_{\geq 0}$. For a hybrid system with no jump dynamics, each solution has a time domain of the form $[0, \infty) \times \{0\}$, $[0, T] \times \{0\}$, or $[0, T) \times \{0\}$ for some $T \geq 0$. No assumptions are made in this entry to guarantee existence of nontrivial solutions since stability theory does not hinge on existence of solutions; rather, it simply makes statements about the behavior of solutions when they exist. To ensure robustness of various stability properties, the following basic regularity assumptions are usually imposed.

Assumption 1 The data (C, F, D, G) satisfy the following conditions:
1. The sets C and D are closed.
2. The set-valued mapping F is outer semi-continuous, locally bounded, and $F(x)$ is nonempty and convex for each $x \in C$.
3. The set-valued mapping G is outer semi-continuous, locally bounded, and $G(x)$ is nonempty for each $x \in D$.

To elaborate further, a set-valued mapping, like F, is said to be *outer semicontinuous* if for each convergent sequence $\{(x_i, y_i)\}_{i=0}^{\infty}$ that satisfies $y_i \in F(x_i)$ for all $i \in \mathbb{Z}_{\geq 0}$, its limit, denoted (x, y), satisfies $y \in F(x)$. It is said to be *locally bounded* if for each bounded set $K_1 \subset \mathbb{R}^n$ there exists a bounded set $K_2 \subset \mathbb{R}^n$ such that, for every $x \in K_1$, every $y \in F(x)$ belongs to K_2; the latter condition is sometimes written $F(K_1) \subset K_2$. If C is closed, f is a function $f : C \to \mathbb{R}^n$ that is continuous, and F is a set-valued mapping that has the single value $f(x)$ for each $x \in C$ and is empty for $x \notin C$, then F is outer semicontinuous, locally bounded, and $F(x)$ is nonempty and convex for each $x \in C$.

Stability Theory

Definitions and Relationships

Given a dynamical system, predicting or controlling the system's long-term behavior is of primary importance. A system's long-term behavior may be more complicated than just converging to an equilibrium point. This fact motivates studying stability of and convergence to a set of points. For simplicity, this entry focuses on stability of sets that are *compact*, that is, they are closed and bounded. A variety of stability concepts are defined below. Each of these concepts applies to continuous-time or discrete-time systems as readily as to hybrid systems.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *Lyapunov stable* for (1) if for each $\varepsilon > 0$ there exists $\delta > 0$ such that for every solution of (1), $x(0, 0) \in \mathcal{A} + \delta \mathbb{B}$ implies $x(t, j) \in \mathcal{A} + \varepsilon \mathbb{B}$ for all $(t, j) \in$

dom x, where $\mathcal{A} + \delta\mathbb{B}$ indicates the set of points whose distance to the set $\mathcal{A}$ is less than or equal to δ. In order for a compact set to be Lyapunov stable for (1), it must be *forward invariant* for (1), that is, each solution of (1) with $x(0,0) \in \mathcal{A}$ satisfies $x(t,j) \in \mathcal{A}$ for all $(t,j) \in$ dom x. However, forward invariance does not necessarily imply Lyapunov stability.

For a compact set $\mathcal{A} \subset \mathbb{R}^n$, its *basin of attraction* for (1), denoted $\mathcal{B}_\mathcal{A}$, is the set of points from which each solution to (1) is bounded and each solution to (1) having an unbounded time domain converges to $\mathcal{A}$, the latter being written mathematically as $\lim_{t+j \to \infty} |x(t,j)|_\mathcal{A} = 0$ where $|x(t,j)|_\mathcal{A}$ denotes the distance of $x(t,j)$ to the set $\mathcal{A}$. Each point that does not belong to $C \cup D$ belongs to $\mathcal{B}_\mathcal{A}$ since there are no solutions from such points. A compact set $\mathcal{A}$ is said to be *attractive* for (1) if its basin of attraction contains a neighborhood of itself, that is, there exists $\varepsilon > 0$ such that $\mathcal{A} + \varepsilon\mathbb{B} \subset \mathcal{B}_\mathcal{A}$. A compact set $\mathcal{A}$ is said to be *globally attractive* if $\mathcal{B}_\mathcal{A} = \mathbb{R}^n$.

A compact set is said to be *asymptotically stable* for (1) if it is Lyapunov stable and attractive for (1). A compact set is said to be *globally asymptotically stable* for (1) if it asymptotically stable for (1) and $\mathcal{B}_\mathcal{A} = \mathbb{R}^n$. It is useful to know that the basin of attraction for an asymptotically stable set is always open.

Theorem 1 *Under Assumption 1, if a compact set is asymptotically stable for (1), then its basin of attraction is an open set.*

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *uniformly attractive* for (1) if it is attractive for (1) and for each compact set $K \subset \mathcal{B}_\mathcal{A}$ and each $\delta > 0$ there exists $T > 0$ such that for every solution x of (1), $x(0,0) \in K$ and $t + j \geq T$ imply $x(t,j) \in \mathcal{A} + \delta\mathbb{B}$. A compact set is said to be *uniformly globally attractive* for (1) if it is globally attractive and uniformly attractive for (1). Uniform attractivity goes beyond attractivity by asking that the amount of time it takes each solution to get close to $\mathcal{A}$ is uniformly bounded over initial conditions in compact subsets of the basin of attraction.

A compact set $\mathcal{A} \subset \mathbb{R}^n$ is said to be *Lagrange stable* relative to an open set $O \supset \mathcal{A}$ for (1) if for each compact set $K_1 \subset O$ there exists a compact set $K_2 \subset O$ such that for every solution of (1), $x(0,0) \in K_1$ implies $x(t,j) \in K_2$ for all $(t,j) \in$ dom x. In Lagrange stability for the case $O = \mathbb{R}^n$, a bound on the initial conditions is given and a bound on the ensuing solutions must be found; this is in contrast to Lyapunov stability where a bound on the solutions is given and a bound on the initial conditions must be found.

A compact set is said to be *uniformly asymptotically stable* for (1) if it is Lyapunov stable, attractive, Lagrange stable relative to its basin of attraction, and uniformly attractive for (1). A compact set is said to be *uniformly globally asymptotically stable* for (1) if it is uniformly asymptotically stable for (1) and $\mathcal{B}_\mathcal{A} = \mathbb{R}^n$. There is no difference between asymptotic stability and uniform asymptotic stability under Assumption 1.

Theorem 2 *Under Assumption 1, a compact set is uniformly asymptotically stable for (1) if and only if it is locally asymptotically stable for (1).*

As noted earlier, forward invariance does not imply Lyapunov stability. However, when coupled with uniform attractivity, Lyapunov stability ensues.

Theorem 3 *Under Assumption 1, a compact set is uniformly asymptotically stable for (1) if and only if it is forward invariant and uniformly attractive for (1).*

Asymptotic stability can be converted to global asymptotic stability by shrinking the flow and jump sets to be compact subsets of the basin of attraction. However, global asymptotic stability of a compact set $\mathcal{A}$ for $x \in C, \dot{x} = f(x)$ for each compact set C does not necessarily imply global asymptotic stability of $\mathcal{A}$ for $\dot{x} = f(x)$.

In some situations it is easier to assert the existence of a compact asymptotically stable set than it is to find one explicitly. In this direction, given a set $X \subset \mathbb{R}^n$, consider the set of points z with the property that there exist a sequence of solutions $\{x_i\}_{i=0}^\infty$ to (1) with initial conditions in X and a sequence of times $\{(t_i, j_i)\}_{i=0}^\infty$ with $(t_i, j_i) \in$ dom x_i for each $i \in \mathbb{Z}_{\geq 0}$ such that $z = \lim_{i \to \infty} x_i(t_i, j_i)$. This set of points is called the ω-limit set of X for (1) and is denoted $\Omega(X)$.

Theorem 4 *Let Assumption 1 hold. For the system (1), if X is compact and $\Omega(X)$ is nonempty and contained in the interior of X (i.e., there exists $\varepsilon > 0$ such that $\Omega(X) + \varepsilon\mathbb{B} \subset X$), then the set $\Omega(X)$ is compact and uniformly asymptotically stable with basin of attraction containing X and equal to the basin of attraction for X.*

Robustness

A given model (C, F, D, G) may have some mismatch with a physical process that it aims to describe. One way to capture some of this mismatch is to consider the behavior of solutions to a system with inflated data $(C_\delta, F_\delta, D_\delta, G_\delta)$, $\delta \geq 0$, defined as follows:

$$C_\delta := \{x \in \mathbb{R}^n : (x + \delta\mathbb{B}) \cap C \neq \varnothing\} \quad (2a)$$

$$F_\delta(x) := \overline{\mathrm{co}}F((x + \delta\mathbb{B}) \cap C) + \delta\mathbb{B} \quad (2b)$$

$$D_\delta := \{x \in \mathbb{R}^n : (x + \delta\mathbb{B}) \cap D \neq \varnothing\} \quad (2c)$$

$$G_\delta := G((x + \delta\mathbb{B}) \cap D) + \delta\mathbb{B}. \quad (2d)$$

The notation $x + \delta\mathbb{B}$ indicates a closed ball of radius δ centered at the point x. Evaluating a set-valued mapping at a set of points means to collect all vectors that belong to the set-valued mapping at any point in the set that serves as the argument of the set-valued mapping. The notation "$\overline{\mathrm{co}}F((x + \delta\mathbb{B}) \cap C)$" indicates the closed, convex hull of the set $\{f \in \mathbb{R}^n : f \in F(z), z \in (x + \delta\mathbb{B}) \cap C\}$. Note that $(C_0, F_0, D_0, G_0) = (C, F, D, G)$. More generally, the components of (C, F, D, G) are contained in $(C_\delta, F_\delta, D_\delta, G_\delta)$. The inflation data in (2) satisfy the regularity properties of Assumption 1 when (C, F, D, G) do.

Proposition 1 *If the data (C, F, D, G) satisfy Assumption 1 then, for each $\delta > 0$, the inflated data $(C_\delta, F_\delta, D_\delta, G_\delta)$ satisfy Assumption 1.*

From the point of view of asymptotic stability, the behavior of solutions to $(C_\delta, F_\delta, D_\delta, G_\delta)$ for $\delta > 0$ small is not too different from those of (C, F, D, G).

Theorem 1 *Under Assumption 1, if $\mathcal{A}$ is asymptotically stable with basin of attraction $\mathcal{B}_\mathcal{A}$ for the hybrid system with data (C, F, D, G), then for*

each $\varepsilon > 0$ and each compact set K satisfying $K \subset \mathcal{B}_\mathcal{A}$, there exist $\delta > 0$ and a compact set $\mathcal{A}_\delta \subset \mathcal{A} + \varepsilon\mathbb{B}$ that is asymptotically stable with $K \subset \mathcal{B}_{\mathcal{A}_\delta}$ for $(C_\delta, F_\delta, D_\delta, G_\delta)$.

The robustness result of Theorem 1 has several consequences beyond the observations in the preceding examples. One of the consequences is the following reduction principle.

Theorem 2 *Under Assumption 1, if $\mathcal{A}_1$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}_1}$ for the hybrid system with data (C, F, D, G) and the compact set $\mathcal{A}_2 \subset \mathcal{A}_1$ is globally asymptotically stable for the hybrid system with data $(C \cap \mathcal{A}_1, F, C \cap \mathcal{A}_2, G)$, then the compact set $\mathcal{A}_2$ is asymptotically stable with basin of attraction $\mathcal{B}_{\mathcal{A}_1}$ for the hybrid system with data (C, F, D, G).*

Lyapunov Functions

Arguably the most common method for establishing asymptotic stability is known as *Lyapunov's method* and uses a Lyapunov function. A function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a *Lyapunov function candidate* for (1) if it is continuously differentiable on an open neighborhood of the flow set C, it is defined for all $x \in C \cup D \cup G(D)$ (dom V denotes the set of points where it is defined), and it is continuous on its domain. Some of these conditions can be relaxed but are imposed in this entry to keep the discussion simple. Given a compact set $\mathcal{A}$ and an open set O satisfying $\mathcal{A} \subset O \subset \mathbb{R}^n$, a Lyapunov function candidate for (1) is called a *Lyapunov function for $(\mathcal{A}, O)$* if:

(L1) For $x \in (C \cup D \cup G(D)) \cap O$, $V(x) = 0$ if and only if $x \in \mathcal{A}$.

(L2) For each $x \in C \cap O$ and $f \in F(x)$, $\langle \nabla V(x), f \rangle \leq 0$.

(L3) For each $x \in D \cap O$ and $g \in G(x)$, $V(g) - V(x) \leq 0$.

A Lyapunov function for $(\mathcal{A}, O)$ is called a *proper Lyapunov function for $(\mathcal{A}, O)$* if, in addition,

(L4) $\lim_{i \to \infty} V(x_i) = \infty$ when the sequence $\{x_i\}_{i=0}^\infty$, satisfying $x_i \in (C \cup D \cup G(D)) \cap O$ for all $i \in \mathbb{Z}_{\geq 0}$, is unbounded or approaches the boundary of O.

The next result does not use Assumption 1, though the rest of the results in this entry do.

Theorem 3 *Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. If there exists a Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is Lyapunov stable for (1). If there exists a proper Lyapunov function for $(\mathcal{A}, O)$ then $\mathcal{A}$ is also Lagrange stable with respect to O for (1).*

We can also conclude asymptotic stability from a Lyapunov function when it is known that there are no complete solutions along which the Lyapunov function is equal to a positive constant.

Theorem 4 *Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. Under Assumption 1, if there exists a Lyapunov function for $(\mathcal{A}, O)$ and there is no solution x of (1) starting in $O \setminus \mathcal{A}$ that has an unbounded time domain and satisfies $V(x(t, j)) = V(x(0, 0))$ for all $(t, j) \in \mathrm{dom}\, x$, then $\mathcal{A}$ is uniformly asymptotically stable for (1). If the Lyapunov function is a proper Lyapunov function for $(\mathcal{A}, O)$, then the basin of attraction for $\mathcal{A}$ contains O.*

The simplest way to rule out solutions that keep a Lyapunov function equal to a positive constant is by finding a *(proper) strict Lyapunov function for* $(\mathcal{A}, O)$, which is a (proper) Lyapunov function for $(\mathcal{A}, O)$ that also satisfies:

(L2$'$) For each $x \in (C \cap O) \setminus \mathcal{A}$ and $f \in F(x)$,
$\langle \nabla V(x), f \rangle < 0$.

(L3$'$) For each $x \in (D \cap O) \setminus \mathcal{A}$ and $g \in G(x)$,
$V(g) - V(x) < 0$.

Theorem 5 *Let $\mathcal{A} \subset O \subset \mathbb{R}^n$ with $\mathcal{A}$ compact and O open. Under Assumption 1, if there exists a strict Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is uniformly asymptotically stable for (1). If there exists a proper strict Lyapunov function for $(\mathcal{A}, O)$, then $\mathcal{A}$ is uniformly asymptotically stable for (1) with basin of attraction containing O.*

While a strict Lyapunov function can be difficult to find, and this fact has motivated other more sophisticated stability analysis tools that have appeared in the literature, it is reassuring to know that whenever $\mathcal{A}$ is compact and asymptotically stable, there exists a proper strict Lyapunov function for $(\mathcal{A}, \mathcal{B}_\mathcal{A})$.

Theorem 6 *Under Assumption 1, if the compact set $\mathcal{A}$ is asymptotically stable for (1), then there exists a proper strict Lyapunov function for $(\mathcal{A}, \mathcal{B}_\mathcal{A})$. More specifically, for each $\lambda > 0$ there exists a smooth function V with $\mathrm{dom}\, V = \mathcal{B}_\mathcal{A}$ that $V(x) = 0$ if and only if $x \in \mathcal{A}$, $\lim_{i \to \infty} V(x_i) = \infty$ when the sequence $\{x_i\}_{i=0}^{\infty}$, satisfying $x_i \in \mathcal{B}_\mathcal{A}$ for all $i \in \mathbb{Z}_{\geq 0}$, is unbounded or tends to the boundary of $\mathcal{B}_\mathcal{A}$, and such that:*
1. *For all $x \in C \cap \mathcal{B}_\mathcal{A}$ and $f \in F(x)$, $\langle \nabla V(x), f \rangle \leq -\lambda V(x)$.*
2. *For all $x \in D \cap \mathcal{B}_\mathcal{A}$ and $g \in G(x)$, $V(g) \leq \exp(-\lambda) V(x)$.*

Summary and Future Directions

Under Assumption 1, stability theory for hybrid dynamical systems is very similar to stability theory for differential equations or difference equations with continuous right-hand sides. In particular, Lyapunov functions are a very common analysis tool for hybrid dynamical systems, though a Lyapunov function can be difficult to find in the same way that they are challenging to find for classical systems. With stability theory for hybrid dynamical systems firmly in place, future research is expected to exploit this theory more fully for the development of control algorithms with new capabilities.

Cross-References

▶ Hybrid Dynamical Systems, Feedback Control of
▶ Lyapunov's Stability Theory

Bibliography

Bainov DD, Simeonov PS (1989) Systems with impulse effect: stability, theory, and applications. Ellis Horwood Limited, Chichester

Branicky MS (1998) Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Trans Autom Control 43:1679–1684

DeCarlo RA, Branicky MS, Pettersson S, Lennartson B (2000) Perspectives and results on the stability and stabilizability of hybrid systems. Proc IEEE 88(7):1069–1082

Goebel R, Sanfelice RG, Teel AR (2009) Hybrid dynamical systems. IEEE Control Syst Mag 29(2):28–93

Goebel R, Sanfelice RG, Teel AR (2012) Hybrid dynamical systems. Princeton University Press, Princeton

Haddad W, Chellaboina V, Nersesov SG (2006) Impulsive and hybrid dynamical systems. Princeton University Press, Princeton

Hespanha JP (2004) Uniform stability of switched linear systems: extensions of LaSalle's invariance principle. IEEE Trans Autom Control 49(4):470–482

Lakshmikantham V, Bainov DD, Simeonov PS (1989) Theory of impulsive differential equations. World Scientific, Singapore/Teaneck

Liberzon D (2003) Switching in systems and control. Birkhauser, Boston

Liberzon D, Morse AS (1999) Basic problems in stability and design of switched systems. IEEE Control Syst Mag 19(5):59–70

Lygeros J, Johansson KH, Simić SN, Zhang J, Sastry SS (2003) Dynamical properties of hybrid automata. IEEE Trans Autom Control 48(1):2–17

Matveev A, Savkin AV (2000) Qualitative theory of hybrid dynamical systems. Birkhauser, Boston

Michel AN, Hou L, Liu D (2008) Stability of dynamical systems: continuous, discontinuous, and discrete systems. Birkhauser, Boston

van der Schaft A, Schumacher H (2000) An introduction to hybrid dynamical systems. Springer, London/New York

Yang T (2001) Impulsive control theory. Springer, Berlin/New York

Stability: Lyapunov, Linear Systems

A. Astolfi

Department of Electrical and Electronic Engineering, Imperial College London, London, UK

Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

Abstract

The notion of stability allows to study the qualitative behavior of dynamical systems. In particular it allows to study the behavior of trajectories close to an equilibrium point or to a motion.

The notion of stability that we discuss has been introduced in 1882 by the Russian mathematician A.M. Lyapunov, in his doctoral thesis; hence, it is often referred to as Lyapunov stability. In this entry we discuss and characterize Lyapunov stability for linear systems.

Keywords

Eigenvalues; Equilibrium points; Linear systems; Motions; Stability

Introduction

Consider a linear, time-invariant, finite-dimensional system, i.e., a system described by equations of the form

$$\begin{aligned}
\sigma x &= Ax + Bu, \\
y &= Cx + Du,
\end{aligned} \tag{1}$$

with $x(t) \in I\!R^n$, $u(t) \in I\!R^m$, $y(t) \in I\!R^p$ and $A \in I\!R^{n \times n}$, $B \in I\!R^{n \times m}$, $C \in I\!R^{p \times n}$, and $D \in I\!R^{p \times m}$ constant matrices. In Eq. (1) $\sigma x(t)$ stands for $\dot{x}(t)$ if the system is continuous-time and for $x(t + 1)$ if the system is discrete-time. Since the system is time-invariant, it is assumed, without loss of generality, that all signals are defined for $t \geq 0$, that is, if the system is continuous-time, then $t \in I\!R^+$, i.e., the set of non-negative real numbers, whereas if the system is discrete-time, then $t \in Z^+$, i.e., the set of non-negative integers. For ease of notation, the argument "t" is dropped whenever this does not cause confusion, and we use the notation $t \geq 0$ to denote either $I\!R^+$ or Z^+. Finally, we use either $x(t, x(0), u)$ or $x(t)$ to denote the solution of the first of equations (1) at a given time $t \geq 0$, with the initial condition $x(0)$ and the input signal u. The former is used when it is important to keep track of the initial state and external input u, whereas the latter is used whenever there is not such a need.

Definition 1 (Equilibrium) Consider the system (1). Assume the input u is constant, i.e., $u(t) = u_0$ for all $t \geq 0$ and for some constant

u_0. A state x_e is an equilibrium of the system associated to the input u_0 if $x_e = x(t, x_e, u_0)$, for all $t \geq 0$.

Proposition 1 (Equilibria of linear systems) *Consider the system ([1](#)) and assume $u(t) = u_0$, for all t, where u_0 is a constant vector. Then the following hold.*

- *If $u_0 = 0$ then the origin is an equilibrium.*
- *For continuous-time systems, if A is invertible, for any u_0 there is a unique equilibrium $x_e = -A^{-1}Bu_0$. If A is not invertible, the system has either infinitely many equilibria or it has no equilibria.*
- *For discrete-time systems, if $I - A$ is invertible, for any u_0 there is a unique equilibrium $x_e = (I - A)^{-1}Bu_0$. If $I - A$ is not invertible, the system has either infinitely many equilibria or it has no equilibria.*

Proposition 2 *Consider the continuous-time, time-invariant, linear system*

$$\dot{x} = Ax + Bu,$$
$$y = Cx + Du,$$

and the initial condition $x(0) = x_0$. Then, for all $t \geq 0$,

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (2)$$

and

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(t). \quad (3)$$

Proposition 3 *Consider the discrete-time, time-invariant, linear system (to simplify the notation we use $x^+(t)$ to denote $x(t + 1)$ and we drop the argument t)*

$$x^+ = Ax + Bu,$$
$$y = Cx + Du,$$

and the initial condition $x(0) = x_0$. Then, for all $t \geq 0$,

$$x(t) = A^t x_0 + \sum_{i=0}^{t-1} A^{t-1-i}Bu(i) \quad (4)$$

and

$$y(t) = CA^t x_0 + \sum_{i=0}^{t-1} CA^{t-1-i}Bu(i) + Du(t). \quad (5)$$

Definitions

In this section we provide some notions and definitions which are applicable to general dynamical systems.

Definition 2 (Lyapunov stability) Consider the system ([1](#)) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0. Let x_e be an equilibrium point. The equilibrium is stable (in the sense of Lyapunov) if for every $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that $\|x(0) - x_e\| < \delta$ implies $\|x(t) - x_e\| < \epsilon$, for all $t \geq 0$, where the notation $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$.

In stability theory the quantity $x(0) - x_e$ is called initial perturbation, and $x(t)$ is called perturbed evolution. Therefore, the definition of stability can be interpreted as follows. An equilibrium point x_e is stable if however we select a *tolerable* deviation ϵ, there exists a (possibly small) neighborhood of the equilibrium x_e such that all initial conditions in this neighborhood yield trajectories which are within the *tolerable* deviation.

The property of stability dictates a condition on the evolution of the system for all $t \geq 0$. Note, however, that in the definition of stability, we have not requested that the perturbed evolution converge asymptotically, that is, for $t \to \infty$, to x_e. This convergence property is very important in applications, as it allows to characterize the situation in which not only the perturbed evolution remains close to the unperturbed evolution, but it also converges to the initial (unperturbed) evolution. To capture this property we introduce a new definition.

Definition 3 (Asymptotic stability) Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0. Let x_e be an equilibrium point. The equilibrium is asymptotically stable if it is stable and if there exists a constant $\delta_a > 0$ such that $\|x(0) - x_e\| < \delta_a$ implies $\lim_{t \to \infty} \|x(t) - x_e\| = 0$.

In summary, an equilibrium point is asymptotically stable if it is stable, and whenever the initial perturbation is inside a certain neighborhood of x_e, the perturbed evolution converges, asymptotically, to the equilibrium point, which is thus said to be attractive. From a physical point of view, this means that all sufficiently small initial perturbations give rise to effects which can be a priori bounded (stability) and which vanish asymptotically (attractivity).

It is important to highlight that, in general, attractivity does not imply stability: it is possible to have an equilibrium of a system which is not stable (i.e., it is unstable), yet for all initial perturbations, the perturbed evolution converges to the equilibrium. This however is not the case for linear systems, as discussed in section "Stability of Linear Systems". We conclude the section with two simple examples illustrating the notions that have been introduced.

Example 1 Consider the discrete-time system $x^+ = -x$, with $x(t) \in I\!R$. This system has a unique equilibrium at $x_e = 0$. Note that for any initial condition $x_0 \in I\!R$, one has

$$x_{2t-1} = -x_0, \qquad x_{2t} = x_0,$$

for all $t \geq 1$ and integer. This implies that the equilibrium is stable, but not attractive.

Example 2 Consider the continuous-time system

$$\dot{x}_1 = \omega x_2, \qquad \dot{x}_2 = -\omega x_1,$$

with ω a positive constant. The system has a unique equilibrium at $x_e = 0$. This equilibrium is stable, but not attractive. To see this note that, along the trajectories of the system, $x_1 \dot{x}_1 + x_2 \dot{x}_2 = 0$, and this implies that, along the trajectories of the system, $x_1^2(t) + x_2^2(t)$ is

constant, i.e., $x_1^2(t) + x_2^2(t) = x_1^2(0) + x_2^2(0)$. Therefore, the state of the system remains on the circle centered at the origin and with radius $\sqrt{x_1^2(0) + x_2^2(0)}$, for all $t \geq 0$: the condition for stability holds with $\delta(\epsilon) = \epsilon$.

Definition 4 (Global asymptotic stability) Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0. Let x_e be an equilibrium point. The equilibrium is globally asymptotically stable if it is stable and if, for all $x(0)$, $\lim_{t \to \infty} \|x(t) - x_e\| = 0$.

The property of (global) asymptotic stability can be strengthened imposing conditions on the convergence speed of $\|x(t) - x_e\|$.

Definition 5 (Exponential stability) Consider the system (1) with $u(t) = u_0$, for all $t \geq 0$ and for some constant u_0. Let x_e be an equilibrium point. The equilibrium is exponentially stable if there exists $\lambda > 0$, in the case of continuous-time systems, and $0 < \lambda < 1$ in the case of discrete-time systems, such that for all $\epsilon > 0$, there exists a $\delta = \delta(\epsilon) > 0$ such that $\|x(0) - x_e\| < \delta$ implies $\|x(t) - x_e\| < \epsilon e^{-\lambda t}$, in the case of continuous-time systems, and $\|x(t) - x_e\| < \epsilon \lambda^t$, in the case of discrete-time systems, for all $t \geq 0$.

Definition 6 (Stability of motion) Consider the system (1). Let

$$\mathcal{M} = \{(t, x(t)) \in T \times I\!R^n\},$$

with $x(t) = x(t, x_0, u)$, for given x_0 and u, and $T = I\!R^+$, in the case of continuous-time systems, and $T = Z^+$, in the case of discrete-time systems, be a motion. The motion is stable if for every $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that $\|x(0) - x_0\| < \delta$ implies

$$\|x(t, x(0), u) - x(t, x_0, u)\| < \epsilon, \qquad (6)$$

for all $t \geq 0$.

The notion of stability of a motion is substantially the same as the notion of stability of an equilibrium. The important issue is that the time-parametrization is important, i.e., a motion is stable if, for small initial perturbations, the

perturbed evolution is close, for any fixed $t \geq 0$, to the non-perturbed evolution. This does not mean that if the perturbed and unperturbed trajectories are close, then the motion is stable: in fact the trajectories may be close but may be followed with different timing, which means that for some $t \geq 0$ condition (6) may be violated.

Stability of Linear Systems

The notion of stability relies on the knowledge of the trajectories of the system. As a result, even if this notion is very elegant and useful in applications, it is in general hard to assess stability of an equilibrium or of a motion. There are, however, classes of systems for which it is possible to give stability conditions without relying upon the knowledge of the trajectories. Linear systems belong to one such class. In this section we study the stability properties of linear systems, and we show that, because of the linear structure, it is possible to assess the properties of stability and attractivity in a simple way. To begin with, we recall some properties of linear systems.

Proposition 4 *Consider a linear, time-invariant system. (Asymptotic) stability of one motion implies (asymptotic) stability of all motions. In particular, (asymptotic) stability of any motion implies and is implied by (asymptotic) stability of the equilibrium $x_e = 0$.*

The above statement, together with the result in Proposition 1, implies the following important properties.

Proposition 5 *If the origin of a linear system is asymptotically stable, then, necessarily, the origin is the only equilibrium of the system for $u = 0$. Moreover, asymptotic stability of the zero equilibrium is always global. Finally, asymptotic stability implies exponential stability.*

The above discussion shows that the stability properties of a motion (e.g., an equilibrium) of a linear system are inherited by all motions of the system. Moreover, for linear systems, local properties are always global properties. This means

that, with some abuse of terminology, we can refer the stability properties to the linear system, for example, we say that a linear system is stable to mean that all its motions are stable. Stability properties of a linear, time-invariant system are therefore properties of the *free* evolution of its state: for this class of systems, it is possible to obtain simple stability tests.

Proposition 6 *A linear, time-invariant system is stable if and only if $\|e^{At}\| \leq k$, for continuous-time systems, or $\|A^t\| \leq k$, for discrete-time systems, for all $t \geq 0$ and for some $k > 0$. It is asymptotically stable if and only if $\lim_{t \to \infty} e^{At} = 0$, for continuous-time systems, or $\lim_{t \to \infty} A^t = 0$, for discrete-time systems. To state the next result we need to define the geometric multiplicity of an eigenvalue. To this end we recall a few facts. Consider a matrix $A \in I\!R^{n \times n}$ and a polynomial $p(\lambda)$. The polynomial $p(\lambda)$ is a zeroing polynomial for A if $p(A) = 0$. Note that, by Cayley-Hamilton Theorem, the characteristic polynomial of A is a zeroing polynomial for A. Among all zeroing polynomials there is a unique monic polynomial $p_M(\lambda)$ with smallest degree. This polynomial is called the minimal polynomial of A. Note that the minimal polynomial of A is a divisor of the characteristic polynomial of A. If A has $r \leq n$ distinct eigenvalues $\lambda_1, \ldots, \lambda_r$, then*

$$p_M(\lambda) = (\lambda - \lambda_1)^{m_1}(\lambda - \lambda_2)^{m_2} \cdots (\lambda - \lambda_r)^{m_r},$$

where the number m_i denotes, by definition, the geometric multiplicity of λ_i, for $i = 1, \cdots, r$. This means that the geometric multiplicity of λ_i equals the multiplicity of λ_i as a root of $p_M(\lambda)$. Recall, finally, that the multiplicity of λ_i as a root of the characteristic polynomial is called algebraic multiplicity.

Proposition 7 *The equilibrium $x_e = 0$ of a linear, time-invariant system is stable if and only if the following conditions hold.*

- *In the case of continuous-time systems, the eigenvalues of A with geometric multiplicity equal to one have non-positive real part, and the eigenvalues of A with geometric multiplicity larger than one have negative real part.*

- *In the case of discrete-time systems, the eigenvalues of A with geometric multiplicity equal to one have modulo not larger than one, and the eigenvalues of A with geometric multiplicity larger than one have modulo smaller than one.*

Proof   Let $\lambda_1, \lambda_2, \cdots, \lambda_r$, with $r \geq 1$, be the distinct eigenvalues of A, i.e., the distinct roots of the characteristic polynomial of A. Then

$$e^{At} = \sum_{i=1}^{r} \sum_{k=1}^{m_i} R_{ik} \frac{t^{k-1}}{(k-1)!} e^{\lambda_i t},$$

for some matrices R_{ik}, where m_i is the geometric multiplicity of the eigenvalue λ_i. This matrix is bounded if and only if the conditions in the statement hold. Similarly,

$$A^t = \sum_{i=1}^{r} \sum_{k=1}^{m_i} R_{ik} \frac{t^{k-1}}{(k-1)!} \lambda_i^{t-k+1},$$

for some matrices R_{ik}, and this is bounded if and only if the conditions in the statement hold.   ◁

Proposition 8 *The equilibrium $x_e = 0$ of a linear, time-invariant system is asymptotically stable if and only if the following conditions hold.*
- *In the case of continuous-time systems, the eigenvalues of A have negative real part.*
- *In the case of discrete-time systems, the eigenvalues of A have modulo smaller than one.*

Proof   The proof is similar to the one of the previous proposition, once it is noted that, for the considered class of systems and as stated in Proposition 6, asymptotic stability implies and is implied by boundedness and convergence of e^{At} or A^t.   ◁

Remark 11.1 For linear, time-varying systems, i.e., systems described by equations of the form

$$\sigma x = A(t)x + B(t)u,$$
$$y = C(t)x + D(t)u,$$

it is possible to provide stability conditions in the spirit of the boundedness and convergence conditions in Proposition 6. These require the definition of a matrix, the so-called monodromy matrix, which describes the free evolution of the state of the system. It is, however, not possible to provide conditions in terms of eigenvalues of the matrix $A(t)$ similar to the conditions in Propositions 7 and 8.

We conclude this discussion with an alternative characterization of asymptotic stability in terms of linear matrix inequalities.

Proposition 9 *The equilibrium $x_e = 0$ of a linear, time-invariant system is asymptotically stable if and only if the following conditions hold.*
- *In the case of continuous-time systems, there exists a symmetric positive definite matrix $P = P'$ such that $A'P + PA < 0$.*
- *In the case of discrete-time systems, there exists a symmetric positive definite matrix $P = P'$ such that $A'PA - P < 0$.*

To complete our discussion we stress that stability properties are invariant with respect to changes in coordinates in the state space.

Corollary 1 *Consider a linear, time-invariant system and assume it is (asymptotically) stable. Then any representation obtained by means of a change of coordinates of the form $x(t) = L\hat{x}(t)$, with L constant and invertible, is (asymptotically) stable.*

Proof   The proof is based on the observation that the change of coordinates transforms the matrix A into $\tilde{A} = L^{-1}AL$ and that the matrices A and $\tilde{A}$ are similar, that is, they have the same characteristic and minimal polynomials.   ◁

Summary and Future Directions

The property of Lyapunov stability is instrumental to characterize the qualitative behavior of dynamical systems. For linear, time-invariant systems, this property can be studied on the basis of

the location, and multiplicity, of the eigenvalues of the matrix A. The property of Lyapunov stability can be studied for more general classes of systems, including nonlinear systems, distributed parameter systems, and hybrid systems, to which the basic definitions given in this article apply.

Cross-References

Recommended Reading

Classical references on Lyapunov stability theory and on stability theory for linear systems are given below.

Bibliography

Antsaklis PJ, Michel AN (2007) A linear systems primer. Birkhäuser, Boston
Brockett RW (1970) Finite dimensional linear systems. Wiley, London
Hahn W (1967) Stability of motion. Springer, New York
Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice-Hall, Upper Saddle River
Lyapunov AM (1992) The general problem of the stability of motion. Taylor & Francis, London
Trentelman HL, Stoorvogel AA, Hautus MLJ (2001) Control theory for linear systems. Springer, London
Zadeh LA, Desoer CA (1963) Linear system theory. McGraw-Hill, New York

State Estimation for Batch Processes

Wolfgang Mauntz
Fakultät Bio- und Chemieingenieurwesen, Technische Universität Dortmund, Dortmund, Germany

Abstract

The information about certain safety or quality parameters during a batch process is valuable for a variety of reasons. In case a direct measurement is too expensive, too slow or nonexisting, a state estimator estimating the desired quantities based on a model and various other measurements may be a good alternative. The most prominent method is calorimetry, where the heat of reaction is measured. This entry gives an overview of different alternatives that support a safe and successful batch operation.

Keywords

Calorimetry; Observer; Soft sensor; State estimator

Introduction

Continuous processes are used to produce a product at a constant rate. They are designed to operate at constant conditions, i.e., the state of the process (conversion, temperatures, pressures, concentrations, etc.) does not vary. In contrast, (semi-)batch processes execute a recipe which means that they are typically operated within a wide range of states. The state of the (semi-batch) process should constantly be monitored. This information is useful for several purposes:

- *Process safety:* abnormal process states such as the accumulation of hazardous substances or reactive materials may lead to dangerous situations such as runaway reactions. The earlier an abnormal state is detected, the better it can be corrected, and the higher is the probability that loss can be avoided.

- *Quality:* if the batch is not operated along the standard trajectory, off-spec product may result which in turn results in extra effort and/or second-grade product if this is discovered in time and in a customer complaint if not discovered before delivery.
- *Profit:* the better the state is known, the less conservative the underlying control scheme needs to be and the more the process can be pushed to its limits. This may lead to a higher throughput, less by-products, or less energy consumption. Advanced control schemes which are typically applied for this purpose require knowledge of the state of the process.

The literature offers a wide range of ways to monitor a batch process. In some processes, the observation of simple measurements like temperatures, pressures, and the time that a process step takes for execution is sufficient to guarantee for safe standard product in minimum time. Examples include some melt-polymerizations.

However, as soon as the process is more complex, more information than just temperatures and pressures is required to monitor the process to meet the goals mentioned above. It may be sufficient to measure other easy to measure properties like conductivities, flow rates, pH values, sound velocities, attenuations, etc. However, in many cases these measurements do not give the complete state of the system. Properties like complex gas phase compositions cannot be measured this way. This might require the installation of more sophisticated measurements as, e.g., NIR spectroscopy, online gas chromatography, Raman spectroscopy, or ion mobility spectroscopy. These measurements require significant effort in terms of installation cost and maintenance. In other situations, no online measurement may be available at all. These cases include the measurement of the distribution of the molecular weight in a polymer melt.

In these cases, where direct online measurements are either too expensive or not available at all, several methods are available to obtain information on the status of the batch (▶ Estimation, Survey on).

- *Statistical Methods*
  Experiences from historical batches are used in a statistical way to predict whether a batch runs normally. This can, e.g., be accomplished by defining a golden batch and a corresponding corridor around these trajectories. More sophisticated methods use principal component analysis (PCA) or partial least squares (PLS) to get a hint at abnormal situations. These methods are even capable of pointing at the origin of a possible problem. They are restricted to problem detection and typically cannot be used for control purposes.
- *Model-Based State Estimation*
  The state of the system (temperatures, pressures, concentrations, etc.) is estimated online which allows for problem detection as well as control applications. This method will be described in more detail in the next chapter.

General reviews of state estimation techniques can be found in Besancon (2007), Schei (2008), and a review of industrial applications is, e.g., given in Fortuna et al. (2007).

Model-Based State Estimation

The basic idea of a state estimator (which is frequently also called *observer* or *soft sensor*) is to run a mathematical model of the process in parallel to the process itself, to compare the available measurements to the values which are predicted by the model, and to correct the estimated state by a suitable function of the observed error, usually an additive correction term that depends on the error. For a state estimator to converge to the true state, the considered system needs to be observable. For details, see ▶ Controllability and Observability. The scheme of a state estimator is sketched in Fig. 1. The real system processes the input $\mathbf{u}$ to give the system state $\mathbf{x}$ which is affected by the system noise $\boldsymbol{\xi}$. The measurements $\mathbf{y}$ are perturbed by the measurement noise $\boldsymbol{\varphi}$. The model predicts a system state $\hat{\mathbf{x}}$ and a measurement $\hat{\mathbf{y}}$. The difference between the measured value $\mathbf{y}$ and predicted value $\hat{\mathbf{y}}$ is then fed back to correct the estimated state.

State Estimation for Batch Processes, Fig. 1 Principle of a state estimator
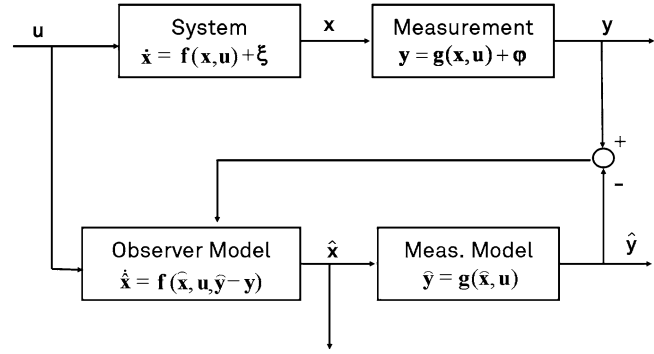
For **linear systems**, the most commonly used state estimators are the *Luenberger observer* and the *Kalman filter* (▶ Kalman Filters). Both multiply the prediction error $(\mathbf{y} - \hat{\mathbf{y}})$ by a weighting matrix $\mathbf{K}$ to update the estimated state $\hat{\mathbf{x}}$:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{y} - \hat{\mathbf{y}})$$

The two techniques use different approaches for determining the matrix $\mathbf{K}$:

Luenberger Observer The basic assumption is that the deviation $\mathbf{e}(t)$ between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is due to wrong initial values $\hat{\mathbf{x}}_0$. $\mathbf{K}$ is computed by choosing the desired speed of convergence of the error

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t)$$
$$= (\mathbf{A} - \mathbf{K}\mathbf{C})\,\mathbf{e}(t)$$

to zero. This is done by placing the eigenvalues of the matrix $(\mathbf{A} - \mathbf{K}\mathbf{C})$ in the left half plane.

Kalman Filter The basic assumption is that the error $\mathbf{e}(t)$ is caused by white noise in the system ξ as well as in the measurement φ. The idea is to minimize the expectation of the quadratic error

$$\min_{\hat{\mathbf{x}}} \quad E\left((\hat{\mathbf{x}}(t) - \mathbf{x}(t))^T (\hat{\mathbf{x}}(t) - \mathbf{x}(t))\right).$$

$\mathbf{K}$ is computed from the noise covariance matrices and the system dynamics and varies with time.

The tuning of the state estimators is not trivial. The larger the absolute value of the eigenvalues in the Luenberger approach, the faster the error will converge to zero but the more prone the state estimator will be to measurement noise. A similar trade-off exists for the Kalman filter where the covariance matrices of the noise terms ξ and φ and the covariance of the initial state ξ_0 need to be defined.

For **nonlinear systems**, a variety of approaches is available. The most frequently used estimators are based on using the nonlinear model for the prediction of the state and linearizations of the system dynamics are used to update the matrix $\mathbf{K}$. The *extended Kalman filter (EKF)* (▶ Extended Kalman Filters) and the *extended Luenberger observer (ELO)* are representatives of this class of approaches. The EKF is most widely used. Extensions are the *constrained EKF* and the *unscented EKF*.

As examples are known where the EKF fails due the nonlinearity of the system, methods based on ideas other than the linearization of system dynamics have been developed. These methods include the *moving horizon estimator* (MHE) (▶ Moving Horizon Estimation) and the *particle filter*. Because of the increasing capabilities of modern computers and significant improvements in dynamic optimization algorithms, the MHE is a very promising alternative. The idea of the method is to minimize the sum of the squared errors of the system noise ξ_l, the measurement noise φ_l, and the error of the initial state ξ_{k-N} which are weighted by weighing matrices $\mathbf{P}_k$, $\mathbf{Q}$ and $\mathbf{R}$ over a predefined horizon of past sampling steps $k - N, \ldots, k$

$$\min_{\xi_i,\varphi_j} \quad \xi_{k-N}^T \mathbf{P}_k^{-1}\xi_{k-N} \; + \; \sum_{l=k-N+1}^{k-1} \xi_l^T \mathbf{Q}^{-1}\xi_l$$
$$+ \sum_{l=k-N+1}^{k} \varphi_l^T \mathbf{R}^{-1}\varphi_l$$

$s.t.$ the system model and the measurement equations are satisfied and further inequality constraints (e.g., physical limits of variables) hold.

The possibility to define constraints on the estimated states, e.g., that concentrations must be nonnegative, is an important advantage of the MHE approach. If the horizon is reduced to one single measurement, the constrained extended Kalman filter results which combines the simplicity of the EKF with the possibility to include constraints on the estimated states. Efficient implementations of the MHE have led to the method being capable of estimating the state of rather large systems in real time (Diehl et al. 2006; Küpper and Engell 2007).

Calorimetry

Temperature measurements are probably the cheapest available measurements in chemical processes, and most plants are typically well equipped with temperature sensors. To exploit temperature measurements, e.g., for the observation of exothermic or endothermic reactions, heat balances are set up and solved for the heat of reaction which then enables the computation of the reaction rate. This is typically referred to as **calorimetry**. Reviews are given, e.g., in Hergeth (2006), McKenna et al. (2000), and Landau (1996). For a jacketed reactor, the heat balance around a semi-batch reactor typically reads (see also Fig. 2)

$$C_{P,R}\frac{dT_R}{dt} = \dot{Q}_R + kA(T_J - T_R)$$
$$+ \sum_i \dot{m}_{F,i} c_{p,Fi}(T_{F,i} - T_R), \quad (1)$$

where $\dot{Q}_R$ represents the heat of reaction, kA the overall heat transfer coefficient between the reactor content and the jacket, T_R the reactor temperature, T_J the jacket temperature, T_F the feed temperature, $C_{P,R}$ the overall heat capacity of the reactor, and the last term on the right side is the enthalpy added by the feed to the reactor. If kA is known, $\dot{Q}_R$ can directly be computed as all other quantities in Eq. (1) are known or measured. This is referred to as *heat flow calorimetry*.

In industrial practice, kA usually is not known and varies over time due to changes of the filling level, changes of the viscosity of the reaction mixture, and fouling. Then other heat balances and measurements can be added to enable a direct computation or estimation of kA. Typically, the jacket heat balance is chosen

$$C_{P,J}\frac{dT_J}{dt} = kA(T_R - T_J) + kA_{\text{jack}}(T_{\text{env}} - T_J)$$
$$+ \dot{m}_J\, c_{p,J}\,(T_{J,in} - T_J). \quad (2)$$

If necessary, also other phenomena like direct heat losses from the reactor content to the environment or the influence of the reactor lid can be taken into account by adding additional terms or additional heat balances. This method is called *heat balance calorimetry*.

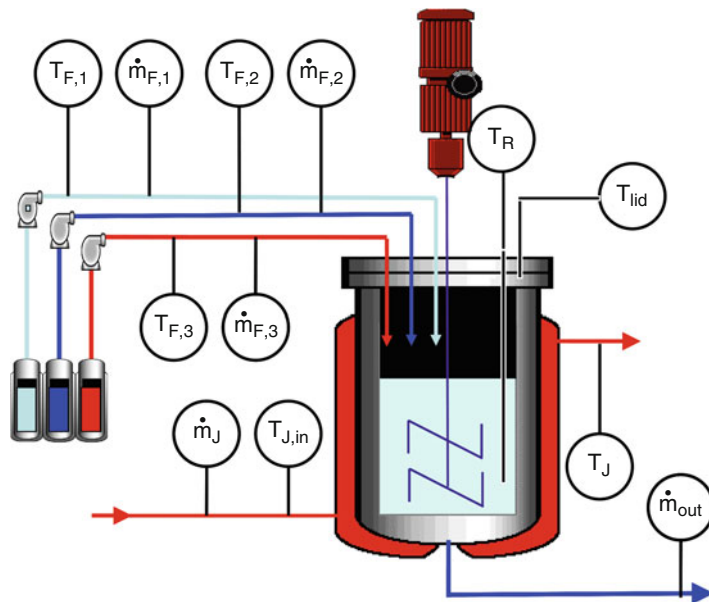In order to compute $\dot{Q}_R$ and kA from Eqs. (1) and (2), two different approaches can be used:
1. Equations (1) and (2) are solved to give

$$(3a)$$
$$\widehat{kA} = \frac{C_{P,J}\frac{dT_J}{dt} - kA_{\text{jack}}(T_{\text{env}} - T_J) - \dot{m}_J\, c_{p,J}\,(T_{J,in} - T_J)}{T_R - T_J}$$

$$\hat{\dot{Q}}_R = C_{P,R}\frac{dT_R}{dt} - kA(T_J - T_R) - \sum_i \dot{m}_{F,i} c_{p,Fi}(T_{F,i} - T_R). \quad (3b)$$

**State Estimation for
Batch Processes, Fig. 2**
The reactor and its jacket
as considered for
calorimetry

In this approach, the derivatives need to be computed from the measurements which introduces noise in the evaluation and requires a filtering either of the derivatives or of the estimates.

2. Equations (1) and (2) are implemented in a nonlinear state estimator. To estimate the unknown quantities $\widehat{kA}$ and $\hat{\dot{Q}}_R$ by this approach, additional assumptions about their dynamics must be made. A common approach is to add the so-called dummy derivatives

$$\frac{d\,\hat{\dot{Q}}_R}{dt} = 0$$
$$\frac{d\,\widehat{kA}}{dt} = 0,$$

The tuning of calorimetric estimation schemes has been discussed in the literature, but for each case, tests in simulation runs using recorded batch data should be performed.

Experimental results of the application of the direct solution equations (3) and an EKF for the estimation of $\dot{Q}_R$ and kA are shown in Fig. 3. A laboratory-scale 10 l metal reactor was filled with water. Cold water was injected into the reactor to simulate the feed of reactants. The reactor is

equipped with a heating rod by which different values of $\dot{Q}_R$ could be simulated. Figure 3a shows the measured temperatures and the feed stream; Fig. 3b shows the estimates. The dotted line displays the measured power uptake, the thin, black line represents the estimates from the evaluation of Eqs. (3), and the gray line shows the results obtained with an EKF. The EKF was tuned slightly more aggressively than the PT1-filter that was used to filter the values of $\hat{\dot{Q}}_R$ and $\widehat{kA}$ that were obtained from Eqs. (3).

It can be seen that the quality of both evaluation methods is comparable. A difference in performance can be seen in the estimation of kA at the points in time where $T_R \approx T_J$. This is due to the denominator in Eq. (3b) which becomes ≈ 0. At this point, kA is unobservable. The EKF estimates of kA are more smooth. This does not have an impact on the estimation of $\dot{Q}_R$ because the heat transfer from the jacket to the reactor is zero at this point. This behavior is of importance if $\widehat{kA}$ is used in other algorithms, e.g., for control purposes.

A practical problem is the determination of the parameters of the system model. Especially the heat capacity of the reactor $C_{P,R}$ is difficult to determine as it is not clear how much impact the reactor material has. Also the heat capacities of
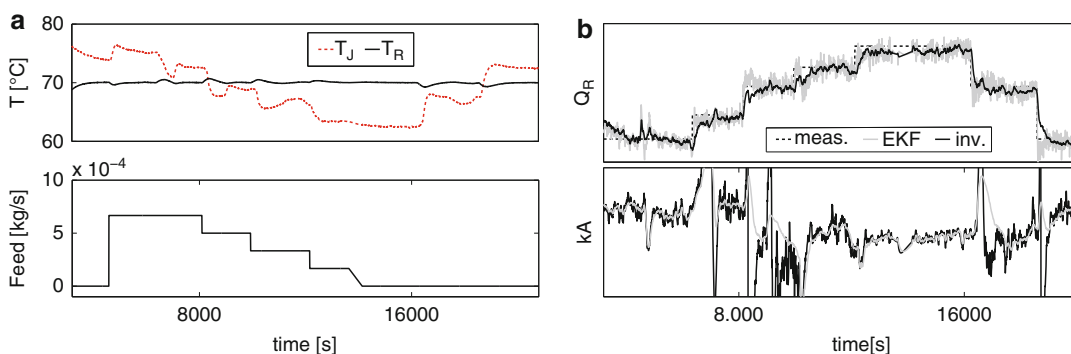
State Estimation for Batch Processes, Fig. 3 Illustration of the results of direct estimation (Eqs. 3) and the use of an EKF. (**a**) Measured data. (**b**) Estimates from inverted equations (3) and EKF as well as measured $\hat{Q}_R$

intermediate products and mixtures with the raw materials and final products may not be known. That is why typically $C_{P,R}$ is considered a "free" parameter which is used to fit the estimates to measured data. If the adjustment of the available parameters is not sufficient to yield a satisfactory performance of the estimator, further extensions can be considered:

- If pressurized vessels are considered, the wall thickness may be considerable, and the heat accumulation may influence the results. In this case, the extension of the set of equations by an equation for the heat transfer through the wall may be considered (Saenz de Buruaga et al. 1997).

- If large-scale vessels are considered, the cooling fluid in the jacket may not be perfectly mixed, and a temperature gradient will be present. In many cases, cooling coils are welded on the outside surface of the reactor. In this case, the equation for the perfectly mixed jacket (Eq. (2)) should be replaced by a model for a plug flow reactor (Krämer and Gesthuisen 2005).

- For large industrial reactors, the perfect mixing assumption of the reactor contents does not necessarily hold true. Especially if polymerization reactions are considered, the reactor content may become rather viscous. A straightforward method to cope with this problem is a detailed computational fluid dynamics (CFD) simulation. However, due

to the numerical complexity, this appears infeasible for online applications. A practical alternative is the placement of several temperature sensors and using a weighted average over their readings. A different approach is the usage of a multi-zonal model, the idea of which resembles the idea of a CFD model; however the number of zones (elements) is much smaller (Bezzo et al. 2004).

Heat balance calorimetry becomes inaccurate if the mass flow through the jacket is so large that the temperature difference between the cooling stream entering the jacket and leaving the jacket $(T_{J,in} - T_J)$ is in the order of magnitude of the measurement error. This mode of operation is typically used in laboratory-scale reactors to avoid temperature gradients in the jacket. To estimate the states in such setups, a technique called *temperature oscillation calorimetry* (TOC) can be used. The idea is to add a small but well-measurable sinusoidal signal to the typically constant set point of the reactor temperature T_R (see Fig. 4 for an example). The reaction of the jacket temperature to the oscillating reactor temperature can be used to compute kA, e.g., by estimating its amplitude δT_J (Tietze et al. 1996) or by adding an additional equation which describes the second derivative of the reactor temperature $\frac{d^2 T_R}{dt^2}$ to the set of heat balances (Mauntz et al. 2007).

Calorimetry estimates the total heat of the reactions in the reactor. It can be used to estimate
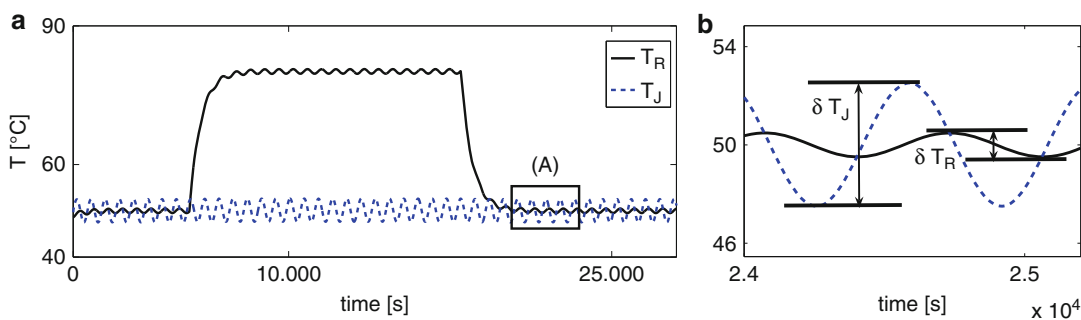
State Estimation for Batch Processes, Fig. 4 Example experiment where TOC is applied. (**a**) Complete example. (**b**) Zoom of rectangle (A)

the overall chemical conversion of a process. Due to its integral character, the heat of reaction of parallel and consecutive reactions cannot be estimated separately (Hergeth 2006). However, if models of the chemical kinetics are known and reliable, it is possible to couple this kinetic model with calorimetry and to observe the complete state of the reaction based on calorimetric estimates. This solution may however not be robust as slight errors in the kinetic model may lead to significant errors in the estimates of all concentrations. In order to build a more robust state estimator, additional measurements should be installed and integrated into the state estimator. For example, for reactions including a phase change from the gas phase to the liquid phase, a pressure measurement may be suitable. For some polymerization reactions, sound velocity and sound attenuation measurements can be valuable (Brandt et al. 2012). The additional measurement can be incorporated into the observation scheme by augmenting the measurement model **g** (see Fig. 1) by the corresponding measurement equation.

Summary

In this contribution, different methods that can be used to determine the states of (semi-)batch reactions have been described. State estimation is useful to reconcile measurement errors and whenever direct online measurements are either too expensive or not available at all.

Linear state estimation is a mature topic. However as chemical batch reactors in most cases have nonlinear dynamics, nonlinear methods should be applied. Extensions of linear state estimators based on linearizations of the system (e.g., the EKF) are the most widely used nonlinear state estimators. However examples are known where these estimators fail. Thus, other approaches, e.g., based on online optimization (MHE), have been developed. They deliver promising results in terms of observation quality and computational speed even for large-scale systems.

The most widespread application of state estimation techniques in batch processes is calorimetry which is suitable for significantly exothermic or endothermic reactions. The heat balances around the reactor contents and the jacket are set up and solved. The estimated heat of reaction is used to estimate the chemical conversion of the process. The method makes use of commonly installed temperature measurements in the reactor. Extensions to include other measurements have been discussed. Problems that typically occur in laboratory-scale reactors can be overcome with the help of temperature oscillation calorimetry.

Cross-References

▶ Control and Optimization of Batch Processes
▶ Controllability and Observability
▶ Estimation, Survey on
▶ Extended Kalman Filters

Bibliography

Besancon G (2007) Nonlinear observers and applications. Springer, Berlin/New York

Bezzo F, Macchietto S, Pantelides CC (2004) A general methodology for hybrid multizonal/CFD models, part I. Theoretical framework AND part II. Automatic zoning. Comput Chem Eng 28:501–525

Brandt H, Sühling D, Engell S (2012) Monitoring emulsion polymerization processes by means of ultra-sound velocity measurements. In: AIChE annual meeting, Pittsburgh, Oct 28–Nov 2

Diehl M, Kühl P, Bock HG, Schlöder JP, Mahn B, Kallrath J (2006) Combined nonlinear MPC and MHE for a copolymerization process. In: 16th European symposium on computer aided process engineering, Garmisch-Patenkirchen, Germany, July 10–13, pp 1527–1532

Fortuna L, Graziani S, Rizzo A, Xibilia MG (2007) Soft sensors for monitoring and control of industrial processes. Springer, London

Hergeth WD (2006) On-line monitoring of chemical reactions. Ullmann's encyclopedia of industrial chemistry. 7th online edn. Wiley-VCH, Weinheim

Küpper A, Engell S (2007) Optimizing control of the hashimoto smb process: Experimental application. In: 8th international IFAC symposium on dynamics and control of process control, Cancun, 6–8 June 2007

Krämer S, Gesthuisen R (2005) Simultaneous estimation of the heat of reaction and the heat transfer coefficient by calorimetry: estimation problems due to model simplification and high jacket flow rates – theoretical development. Chem Eng Sci 60:4233–4248

Landau RN (1996) Expanding the role of reaction calorimetry. Thermochim Acta 289:101–126

Mauntz W, Diehl M, Engell S (2007) Moving horizon estimation and optimal excitation in temperature oscillation calorimetry. In: DYCOPS, Cancun, 6–8 June 2007

McKenna TF, Othman S, Févotte G, Santos AM, Hammouri H (2000) An integrated approach to polymer reaction engineering: a review of calorimetry and state estimation. Polym React Eng 8(1):1–38

Saenz de Buruaga I, Armitage PD, Leiza JR, Asua JM (1997) Nonlinear control for maximum production rate of latexes of well-defined polymer composition. Ind Eng Chem Res 36:4243–4254

Schei TS (2008) On-line estimation for process control and optimization applications. J Process Control 18:821–828

Tietze A, Lüdke I, Reichert K-H (1996) Temperature oscillation calorimetry in stirred tank reactors. Chem Eng Sci 51(11):3131–3137

Statistical Process Control in Manufacturing

O. Arda Vanli[1] and Enrique Del Castillo[2]
[1]Department of Industrial and Manufacturing Engineering, High Performance Materials Institute Florida A&M University and Florida State University, Tallahassee, FL, USA
[2]Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA, USA

Abstract

Statistical process control has been successfully utilized for process monitoring and variation reduction in manufacturing applications. This entry aims to review some of the important monitoring methods. Topics discussed include: Shewhart's model, $\bar{X}$ and R control charts, EWMA and CUSUM charts for monitoring small process shifts, process monitoring for autocorrelated data, and integration of statistical and engineering (or automatic) control techniques. The goal is to provide readers from control theory, mechanical engineering, and electrical engineering an expository overview of the key topics in statistical process control.

Keywords

CUSUM; EWMA; Feedback control; Shewhart control chart; Time-series analysis

Introduction

Variation control is an important goal in manufacturing. The main set of tools for variation control used in discrete-part manufacturing industries up to the 1960s was developed by W. Shewhart in the 1920s and is known today as statistical process control, or SPC (Shewhart 1939). Shewhart's SPC model assumes that the process varies about a fixed mean and that consecutive observations from a process are independent, as follows:

$$Y_t = \mu_0 + \epsilon_t \qquad (1)$$

in which μ_0 is the in-control process mean and ϵ_t is iid (independent identically distributed) white noise $\epsilon \overset{\text{iid}}{\sim} N(0, \sigma^2)$. The Shewhart model can be used in distinguishing assignable cause variation from common cause variation. For example, a mean change from μ_0 to $\mu_1 = \mu_0 + \delta$ (where δ is the unknown magnitude of change) or a variance increase from σ_0^2 to σ_1^2 at an unknown point in time can be detected as assignable causes.

The objective of this entry is to highlight some of the important references in the SPC literature and to discuss similarities and joint applications SPC has with automatic process control. The literature on statistical process control and applications to engineering problems is vast; therefore, no effort is made for an exhaustive review. More complete reviews of the literature on statistical process control and adjustment methods can be found in texts including Montgomery (2013), Ryan (2011), and Del Castillo (2002).
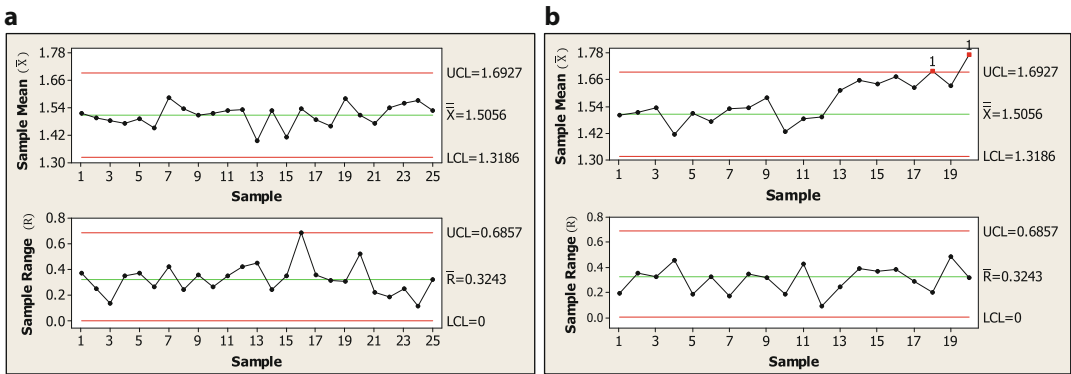
Shewhart Control Charts

Shewhart's $\bar{X}$ and R control charts are used to distinguish between common cause and assignable causes of variation (Shewhart 1939) by monitoring, respectively, the process mean and process variance. "Common cause" variation is the natural variability of the process due to uncontrollable factors in the environment that is not avoidable without substantial changes

to the process. "Assignable cause" variation is due to unwanted disturbances or upsets to the process that can be detected and removed to produce acceptable quality products. When only common cause variation exists, the process is said to be operating "in statistical control." Assignable causes of variation include operator changes, machine calibration errors or raw material variation between suppliers.

Another concept that is closely related to the Shewhart's model is process capability. Process capability indices are used to assess whether the process is operating in a satisfactory manner with respect to the engineering specifications. It is crucial to attain a stable process (eliminating all problematic causes) before undertaking such a capability analysis because only when the samples come from a stable probability distribution can the future behavior of the process be predicted "within probability limits determined by the common cause system" (Box and Kramer 1992).

Figure 1 illustrates the two main phases, referred to as Phase I and Phase II, in constructing Shewhart charts (Sullivan 2002), using semiconductor lithography process data given in Montgomery (2013). It is desired to establish a statistical control of the width of the resist using $\bar{X}$ and R charts. Twenty-five preliminary subgroups, each of size five wafers, were taken at one-hour intervals and the resist width is measured. In Phase I, "retrospective analysis," the historical data from the process is analyzed to bring an initially out-of-control process into

Statistical Process Control in Manufacturing, Fig. 1 Shewhart $\bar{X}$ and R charts from (**a**) Phase I analysis and (**b**) Phase II analysis

statistical control. Subgroups $y_1, \ldots, y_n$ of size n are taken, and subgroup average $\bar{y}$ is used to monitor process mean μ_0, and the subgroup range is used to monitor standard deviation of the process mean $\sigma_{\bar{Y}} = \sigma/\sqrt{n}$. The upper and lower control limits are found for the $\bar{X}$ chart as $\{UCL, LCL\} = \mu_0 \pm L\sigma_{\bar{Y}}$ where L is a constant representing the width of the control limits. Commonly chosen three-sigma limits (i.e., $L = 3$) provide a probability $p = 0.0027$ that a single point falls outside the limits when process is in control ("false alarm probability"). Points that fall outside the control limits are investigated, and if an assignable cause was identified, then this point is omitted and control limits are recalculated. This is repeated until no further points plot outside the limits. In Phase II these charts are used to detect shifts in the process mean and variability.

The $\bar{X}$ and R charts from Phase I data in Fig. 1a indicate statistical control; hence the computed control limits can be used for Phase II monitoring. Twenty additional subgroups (also of size 5) are taken in Phase II while the control charts are in use. The Phase II charts shown in Fig. 1b indicate that process variability is stable but the process mean has shifted at subgroup 18. The general trend in the $\bar{X}$ chart indicates that process mean probably has shifted earlier around subgroup 13.

EWMA, CUSUM, and Changepoint Estimation

Shewhart charts can detect large magnitude process upsets reasonably well; however, they are relatively slow to detect small shifts. In order to reduce the reaction time for smaller shifts, a set of "runs" rules (e.g., two out of three runs beyond 2σ limits or four out of five runs beyond 1σ limits) has been proposed Western Electric (1956). A more systematic method is to accumulate information over successive observations using CUSUM and EWMA statistics rather than basing the detection on a single sample. In the cumulative sum (CUSUM) chart, a running total $\sum_{i=1}^{t}(\bar{Y}_t - \mu_0)$ is plotted against subgroup number t, and a shift from the in-control mean μ_0 is

signaled by an upward or downward linear trend in the plot. A two-sided CUSUM is defined as Woodall and Adams (1993):

$$S_t^{\pm} = \max\{\pm Z_t - k + S_{t-1}^{\pm}, 0\} \text{ for } t = 1, 2, \ldots \tag{2}$$

where S_t^{+} and S_t^{-} are the one-sided upper and lower cusums, respectively, $Z_t = (\bar{Y}_t - \mu_0)/\sigma_{\bar{Y}}$ is the standardized subgroup average, $k = |\mu_1 - \mu_0|/(2\sigma)$ is the reference value, and μ_1 is the level of process mean to be detected. An out-of-control signal is given at the first t for which $S_t > h$ where h is a suitably chosen threshold, usually selected based on the desired average number of samples to signal an alarm, also called the average run length (ARL). The recommended value for the threshold h is 4 or 5 (corresponding to four or five times the process standard deviation σ), and the value for the reference k is almost always taken as 0.5 (corresponding to shift size $|\mu_1 - \mu_0| = \sigma$) (Montgomery 2013).

Another chart that accumulates deviations over several samples is the exponentially weighted moving average (EWMA) which is based on the statistic (Lucas and Saccucci 1990)

$$Z_t = \lambda \bar{Y}_t + (1 - \lambda)Z_{t-1} \tag{3}$$

where $0 < \lambda < 1$ is a smoothing constant. Smaller λ provides large smoothing (similar to a large subgroup size n in the Shewhart charts). The starting value is the in-control mean $Z_0 = \mu_0$. It can be shown that Z_t is a weighted average of all previous sample means, where the weights decrease geometrically with the age of the subgroup mean. The EWMA statistic is plotted against the control limits $\mu_0 \pm L\sigma_{\bar{Y}}\sqrt{(\lambda/(2-\lambda))[1-(1-\lambda)^{2t}]}$. Shewhart charts that are effective for large shifts are more useful for Phase I, and CUSUM or EWMA charts that are effective for small shifts are more appropriate for Phase II.

We illustrate in Fig. 2 how to monitor with CUSUM and EWMA charts with the lithography data. The in-control process mean and standard deviation μ_0 and σ are found from the Phase I data. CUSUM upper and lower statistics $S_t^{\pm}$ computed with Phase II data are plotted in Fig. 2a

S
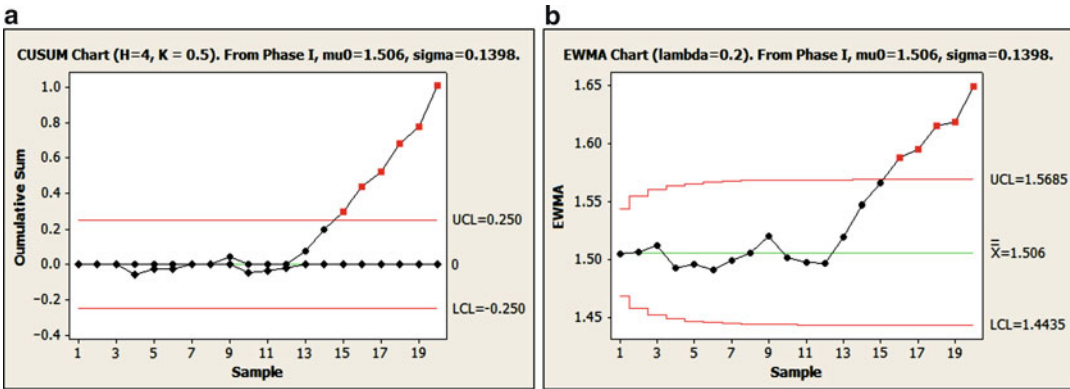
Statistical Process Control in Manufacturing, Fig. 2 Phase II charts for lithography data (**a**) CUSUM chart and (**b**) EWMA chart

(reference value $k = 0.5$ and threshold $h = 4$ are used.). The upper cusum statistic S_t^+ crosses the upper control limit indicating an upward shift at subgroup 15. The EWMA statistic applied with $\lambda = 0.2$ on Phase II data, shown Fig. 2b, crosses the upper control limit at subgroup 16. Both charts have improved the reaction times of the Shewhart chart.

When a control chart signals an assignable cause, it does not indicate when the process change actually occurred. Estimating the instant of the change, or *changepoint* estimation, is especially useful in Phase I analysis where little is known about the process, and it is important to identify and remove the out-of-control samples from consideration (Hawkins et al. 2003; Basseville and Nikiforov 1993; Pignatiello and Samuel 2001). The process is modeled as

$$Y_i \sim N(\mu_1, \sigma^2) \text{ for } i = 1, 2, \dots, \tau$$
$$Y_i \sim N(\mu_2, \sigma^2) \text{ for } i = \tau + 1, \dots, n \quad (4)$$

where τ is the unknown changepoint, at which the in-control mean μ_1 is assumed to shift to a new value μ_2 assuming μ_1, σ are known but μ_2 is unknown. A generalized likelihood ratio (GLR) test statistic $\Lambda_t = \sum_{i=1}^{t} \log f_2(y_i)/f_1(y_i)$ is used to test the hypothesis of a changepoint against the null hypothesis that there is no change. Assuming normality $f(y) = 1/\sqrt{2\pi\sigma} \exp[-(y - \mu)^2/(2\sigma^2)]$ is the probability density function of the quality characteristic. The changepoint model

is equivalent to the CUSUM chart when all parameters μ_1, μ_2 and σ are known a priori. For the lithography Phase II data in Fig. 1b, it can be shown that the changepoint can be estimated as subgroup 13.

SPC on Controlled and Autocorrelated Processes

It is well known that automatic control performance relies heavily on the accuracy of the process models. An active field of research in recent years is the monitoring of controlled systems using SPC charts (Box and Kramer 1992) in order to reduce the effect of model accuracy. Shewhart charts can be used to monitor the output of a feedback-controlled process; however, as the controller effectively corrects the shift, only a short window of opportunity is provided to detect the shift (Vander Wiel et al. 1992). Tsung and Tsui (2008) showed that monitoring the control actions gives better run-length performance than monitoring the output for small- and medium-size shifts, and monitoring the output gives better performance for large shifts. In monitoring controlled processes, measurements taken at short intervals with positive autocorrelation usually inflate the rate of false alarms (Harris and Ross 1991). Widening the control limits and monitoring the residuals of a time-series model fitted to the observations are some of the strategies
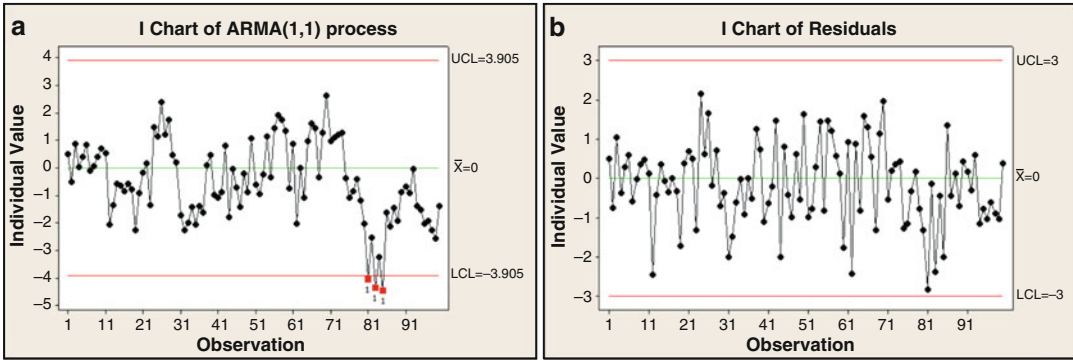
Statistical Process Control in Manufacturing, Fig. 3 (**a**) Shewhart chart for autocorrelated process. (**b**) Shewhart chart for residuals
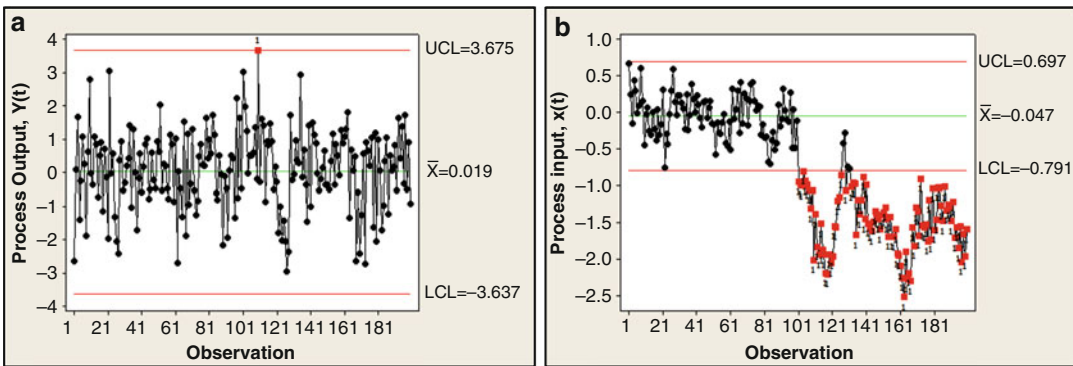
Statistical Process Control in Manufacturing, Fig. 4 (**a**) Shewhart chart for controlled process Y_t. (**b**) Shewhart chart for input X_t

employed to reduce the number of false alarms (Alwan and Roberts 1988).

To illustrate the effects of autocorrelation, we consider simulated data from an autoregressive moving average ARMA(1,1) time-series disturbance process $D_t = 0.8D_{t-1} + \epsilon_t - 0.3\epsilon_{t-1}$ (Box et al. 1994) defined with the white noise process $\epsilon_t \overset{iid}{\sim} N(0, 1^2)$ (with in-control mean $\mu_0 = 0$ and variance $\sigma_D^2 = 1.694$). Figure 3a shows a realization of the process monitored with a Shewhart chart (control limits at $\mu_0 \pm 3\sigma_D$). Due to autocorrelation, false alarms are signaled at samples 81–83. Figure 3b shows the control chart monitoring of the residuals of an ARMA(1,1) model. Residuals (standard normal with mean 0 and variance 1) are not autocorrelated, so the Shewhart chart for residuals does not signal any false alarms.

We illustrate monitoring of controlled processes with simulated data from a transfer function model $Y_t = 2X_{t-1} + D_t$ where X_t are the adjustments made on the process. A proportional integral control rule $X_t = -0.1Y_t - 0.15\sum_{i=1}^{t} Y_i$ is employed, and the disturbance D_t is assumed to follow the ARMA model considered earlier. As an assignable cause, the disturbance mean has shifted at sample 100 by a magnitude of $3\sigma_D$. Figure 4 shows the Shewhart charts monitoring the output Y_t and the input X_t. The effect of assignable cause (at sample 100) on the output is quickly removed by the controller; however, a sustained shift remains in the control input. The control chart for the input Fig. 4b signals the first alarm at sample 101 (much quicker) than the control chart for the output Fig. 4a which signals at sample 110.

Summary and Future Directions

In this entry we reviewed some of the commonly used statistical process monitoring methods for manufacturing systems. Due to space limitations, only several important topics including Phase I and Phase II monitoring with Shewhart, EWMA, and CUSUM charts were discussed, highlighting main applications with numerical examples. Other current research areas include multivariate methods for monitoring processes with multiple quality characteristics taking advantage of relationships among them (Lowry and Montgomery 1992), profile monitoring for processes that generate functional data (Woodall et al. 2004), multistage monitoring for processes with multiple processing steps and variation transmission (Tsung et al. 2008), and run-to-run EWMA control for semiconductor manufacturing processes that require handling of multiple types of products, operators, and machine tools (Butler and Stefani 1994).

Cross-References

▶ Controller Performance Monitoring
▶ Multiscale Multivariate Statistical Process Control
▶ Run-to-Run Control in Semiconductor Manufacturing

Bibliography

Alwan LC, Roberts HV (1988) Time-series modeling for statistical process control. J Bus Econ Stat 6(1):87–95

Basseville ME, Nikiforov IV (1993) Detection of abrupt changes: theory and application. Prentice-Hall, Englewood Cliffs

Box GEP, Kramer T (1992) Statistical process monitoring and feedback adjustment: a discussion. Technometrics 34(3):251–267

Box GEP, Jenkins GW, Reinsel GC (1994) Time series analysis, forecasting and control. Prentice Hall, Englewood Cliffs, NJ

Butler SW, Stefani JA (1994) Supervisory run-to-run control of polysilicon gate etch using in situ ellipsometry. IEEE Trans Semicond Manuf 7(2):193–201

Del Castillo E (2002) Statistical process adjustment for quality control. Wiley, New York

Harris TJ, Ross WH (1991) Statistical process control procedures for correlated observations. Can J Chem Eng 69(1):48–57

Hawkins DM, Peihua Q, Chang WK (2003) The change-point model for statistical process control. J Qual Technol 35(4):355–366

Lowry CA, Montgomery DC (1995) A review of multivariate control charts. IIE Trans 27(6):800–810

Lucas JM, Saccucci MS (1990) Exponentially weighted moving average control schemes: properties and enhancements. Technometrics 32(1):1–12

Montgomery DM (2013) Introduction to statistical quality control. 7th edn. Wiley, New York

Pignatiello JJ, Jr, Samuel TR (2001) Estimation of the change point of a normal process mean in SPC applications. J Qual Technol 33(1):82–95

Ryan TP (2011) Statistical methods for quality improvement, 3rd edn. Wiley, New York

Shewhart WA (1939) Statistical method from the viewpoint of quality control. The Graduate School of the Department of Agriculture, Washington, D.C.

Sullivan JH (2002) Detection of multiple change points from clustering individual observations. J Qual Technol 34(4):371–383

Tsung F, Tsui KL (2003) A mean-shift pattern study on integration of SPC and APC for process monitoring. IIE Trans 35(3):231–242

Tsung F, Li Y, Jin M (2008) Statistical process control for multistage manufacturing and service operations. Int J Serv Oper Inform 3(2):191–204

Vander Wiel SA, Tucker WT, Faltin FW, Doganaksoy N (1992) Algorithmic statistical process control: concepts and an application. Technometrics 34(3):286–297

Western Electric (1956) Statistical Quality Control Handbook, Western Electric Corporation, Indianapolis, IN

Woodall WH, Adams BM (1993) The statistical design of CUSUM charts. Qual Eng 5(4):559–570

Woodall WH, Spitzner DJ, Montgomery DC, Gupta S (2004) Using control charts to monitor process and product quality profiles. J Qual Technol 36(3):309–320

Stochastic Adaptive Control

Tyrone Duncan and Bozenna Pasik-Duncan
Department of Mathematics, University of Kansas, Lawrence, KS, USA

Abstract

Stochastic adaptive control denotes the control of partially known stochastic control systems. The stochastic control systems can be described by discrete- or continuous-time Markov chains

or Markov processes, linear and nonlinear difference equations, and linear and nonlinear stochastic differential equations. The solution of a stochastic adaptive control problem typically requires the identification of the partially known stochastic system and the simultaneous control of the partially known system using the information from the concurrent identification scheme. Two desirable goals for the solution of a stochastic adaptive control problem are called self-tuning and self-optimality. Self-tuning denotes the convergence of the family of adaptive controls indexed by time to the optimal control for the true system. Self-optimizing denotes the convergence of the long-run average costs to the optimal long-run average cost for the true system. Typically to achieve the self-optimality, it is important that the family of parameter estimators from the identification scheme be strongly consistent, that is, this family converges (almost surely) to the true parameter values. Thus, with self-optimality, asymptotically a partially known system can be controlled as well as the corresponding known system.

Keywords

Bayesian estimation; Brownian motion; Markov processes; Self-tuning regulators

Motivation and Background

In almost every formulation of a stochastic control problem from a physical system, the physical system is incompletely known so the stochastic system model is only partially known. This lack of knowledge can often be described by some unknown parameters for a mathematical model, and the noise inputs for the model can describe unmodeled dynamics or perturbations to the system. The lack of knowledge of some parameters of the model can be modeled either by random variables with known prior distributions or as fixed unknown values. The former description requires Bayesian estimation, and the latter description requires parameter estimation such as least squares or maximum likelihood.

Stochastic adaptive control arose as a natural evolution from the results in stochastic control, and in particular it developed for some well-known control problems. The optimal control of Markov chains had been developed for some time, so it was natural to investigate the adaptive control of Markov chains. Mandl (1973) was probably the first to consider this adaptive control problem in generality. His conditions for strong consistency of a family of estimators were fairly restrictive. Borkar and Varaiya (1982) simplified the conditions for the estimation part of the problem by only requiring convergence of the estimators of the parameters so that the resulting transition probabilities of the Markov chain are identical to the transition probabilities for the true optimal solution.

A second major direction for stochastic adaptive control is described by ARMAX (autoregressive-moving average with exogenous inputs) models. These are discrete-time models that can be described in terms of polynomials in a time shift operator. A closely related and often equivalent model is multidimensional linear difference equations in a state-space form. Since the solution of the infinite time horizon stochastic control problem was available in the late 1950s, it was natural to consider the adaptive control problem. Methods such as least squares, weighted least squares, maximum likelihood, and stochastic approximation were used for parameter identification and a certainty equivalence adaptive control for the system, that is, using the current estimate of the parameters as the true parameters to verify self-optimality. An important development in stochastic adaptive control is a result called the self-tuning regulator where the convergence of estimators of unknown parameters implied the convergence of the output tracking error (Astrom and Wittenmark 1973; Goodwin et al. 1981; Guo 1995, 1996; Guo and Chen 1991; Kumar 1990).

A number of monographs treat various aspects of stochastic adaptive control problems, e.g., Astrom and Wittenmark (1989), Chen and Guo (1991), Kumar and Varaiya (1986), and Ljung and Soderstrom (1983). An extensive survey article on the early years of stochastic adaptive control is given by Kumar (1985).

Structures and Approaches

Various requirements can be made for the adaptive control of a stochastic system. It can only be required that the family of adaptive controls is stabilizing the unknown system or that the family of adaptive controls converges to the optimal control for the true system or that the family of adaptive controls has a long-run average cost that is equal to the optimal average cost for the true system. The identification part of the adaptive control problem can be Bayesian estimation (Kumar 1990) if the parameters are assumed to be random variables or parameter estimation (Bercu 1995; Lai and Wei 1982) if the parameters are assumed to be unknown constants. The identification scheme may also incorporate information about the running cost.

For linear systems with white noise inputs, it is well known to use least squares (or equivalently maximum likelihood) estimation to estimate parameters. However, for stochastic adaptive control problems, the sufficient conditions for the family of estimators to be strongly consistent are fairly restrictive (e.g., Lai and Wei 1982), and in fact the family of estimators may not even converge in general. A weighted least squares estimation scheme can guarantee convergence of the family of estimators (Bercu 1995) and can often be strongly consistent (Guo 1996). Some other estimation methods are stochastic approximation (Guo and Chen 1991) and an ordinary differential equation approach (Ljung and Soderstrom 1983). For discrete-time nonlinear systems, a family of strongly consistent estimators may not converge sufficiently rapidly even to stabilize the nonlinear system (Guo 1997).

The study of stochastic adaptive control of continuous-time linear stochastic systems with long-run average quadratic costs developed somewhat after the corresponding discrete-time study (e.g., Duncan and Pasik-Duncan 1990). A solution with basically the natural assumptions from the solution of the known system problem using a weighted least squares identification scheme is given in Duncan et al. (1999).

Another family of stochastic adaptive control problems is described by linear stochastic equations in an infinite dimensional Hilbert space. These models can describe stochastic partial differential equations and stochastic hereditary differential equations. Some linear-quadratic-Gaussian control problems have been solved, and these solutions have been used to solve some corresponding stochastic adaptive control problems (e.g., Duncan et al. 1994a).

Optimal control methods such as Hamilton-Jacobi-Bellman equations and a stochastic maximum principle have been used to solve stochastic control problems described by nonlinear stochastic differential equations (Fleming and Rishel 1975). Thus, it was natural to consider stochastic adaptive control problems for these systems. The results are more limited than the results for linear stochastic systems (e.g., Duncan et al. 1994b).

Other stochastic adaptive control problems have recently emerged that are modeled by multi-agents, such as mean field stochastic adaptive control problems (e.g., Nourian et al. 2012).

A Detailed Example: Adaptive Linear-Quadratic-Gaussian Control

This example is a model that is the most well known continuous-time stochastic adaptive control problem. Likewise for a known continuous-time system, this stochastic control problem is the most basic and well known. The controlled system is described by the following stochastic differential equation:

$$dX(t) = AX(t)dt + BU(t)dt + CdW(t)$$
$$X(0) = X_0$$

where $X(t) \in \mathbb{R}^n, U(t) \in \mathbb{R}^m$, and $(W(t), t \geq 0)$ is an $\mathbb{R}^p$-valued standard Brownian motion and (A, B, C) are appropriate linear transformations. $X(t)$ is the state of the system at time t and $U(t)$ is the control at time t. It is assumed that A, B, C are unknown linear transformations. The cost functional, $J(\cdot)$, is a long-run average (ergodic) quadratic cost functional that is given by

$$J(U) = \lim \sup_{T \to \infty} \frac{1}{T} \int_0^T < QX(t), X(t) >$$
$$+ < RU(t), U(t) > dt$$

where $R > 0$ and $Q \geq 0$ are symmetric linear transformations and $< \cdot, \cdot >$ is the canonical inner product in the appropriate Euclidean space. The standard assumptions for the control of the known system are made also for the adaptive control problem, that is, the pair (A, B) is controllable and $(A, Q^{\frac{1}{2}})$ is observable. An optimal control for the known system is

$$U^0(t) = -R^{-1}B^T S X(t)$$

where S is the unique positive, symmetric solution of the following algebraic Riccati equation:

$$A^T S + SA - SBR^{-1}B^T S + Q = 0$$

The optimal cost is

$$J(U^0) = tr(C^T S C)$$

The unknown quantity $C^T C$ can be identified given $(X(t), t \in [a, b])$ for $a < b$ arbitrary from the quadratic variation of Brownian motion, so the identification of C is not considered here. Since it is assumed that the pair (A, B) is unknown, the system equation is rewritten in the following form:

$$dX(t) = \theta^T \varphi(t)dt + C dW(t)$$

where $\theta^T = [A \; B]$ and $\varphi^T(t) = [X^T(t) \; U^T(t)]$. A family of continuous-time weighted least squares recursive estimators $(\theta(t), t \geq 0)$ of θ is given by the following stochastic equation:

$$d\theta(t) = a(t)P(t)\varphi(t)[dX^T(t) - \varphi^T(t)\theta(t)dt]$$
$$dP(t) = -a(t)P(t)\varphi(t)\varphi^T(t)P(t)dt$$

where $(a(t), t \geq 0)$ is a suitable family of positive stochastic weights (Duncan et al. 1999). A family of estimates $(\hat{\theta}(t), t \geq 0)$ is obtained from $(\theta(t), t \geq 0)$ and is expressed as $\hat{\theta}(t) = [A(t) \; B(t)]$ (Duncan et al. 1999).

A process $(S(t), t \geq 0)$ is obtained using $(A(t), B(t))$ by solving the following stochastic algebraic Riccati equation for each $t \geq 0$:

$$A^T(t)S(t) + S(t)A(t)$$
$$- S(t)B(t)R^{-1}B^T(t)S(t) + Q = 0$$

A certainty equivalence method is used to determine the control, that is, it is assumed that the pair $(A(t), B(t))$ is the correct pair for the true system, so a certainty equivalence adaptive control $U(t)$ is given by

$$U(t) = R^{-1}B^T S(t) X(t)$$

It can be shown (Duncan et al. 1999) that the family of estimators $((A(t), B(t)), t \geq 0)$ is strongly consistent and that the family of adaptive controls given by the previous equality is self-optimizing, that is, the long-run average cost $J(U) = J(U^0) = tr(C^T S C)$ where S is the solution of the algebraic Riccati equation for the true system.

Future Directions

A number of important directions for stochastic adaptive control are easily identified. Only three of them are described briefly here. The adaptive control of the partially observed linear-quadratic-Gaussian control problem (Fleming and Rishel 1975) is a major problem to be solved using the same assumptions of controllability and observability as for the known system. This problem is a generalization of the example given above where the output (linear transformation) of the system is observed with additive noise and the family of controls is restricted to depend only on these observations. Another major direction is to modify the detailed example above by replacing the Brownian motion in the stochastic equation for the state by an arbitrary fractional Brownian motion or by an arbitrary square-integrable stochastic process with continuous sample paths. For this latter problem it is necessary to use recent results for optimal controls for the true

system and to have strongly consistent families of estimators. A third major direction is the adaptive control of nonlinear stochastic systems.

Cross-References

▸ Stochastic Linear-Quadratic Control
▸ System Identification: An Overview

Bibliography

Astrom KJ, Wittenmark B (1973) On self-tuning regulators. Automatica 9:185–199

Astrom KJ, Wittenmark B (1989) Adaptive control. Addison-Wesley, Reading

Bercu B (1995) Weighted estimation and tracking for ARMAX models. SIAM J Control Optim 33:89–106

Borkar V, Varaiya P (1982) Identification and adaptive control of Markov chains. SIAM J Control Optim 20:470–489

Chen HF, Guo L (1991) Identification and stochastic adaptive control. Birkhauser, Boston

Duncan TE, Pasik-Duncan B (1990) Adaptive control of continuous time linear systems. Math Control Signals Syst 3:43–60

Duncan TE, Maslowski B, Pasik-Duncan B (1994a) Adaptive boundary and point control of linear stochastic distributed parameter systems. SIAM J Control Optim 32:648–672

Duncan TE, Pasik-Duncan B, Stettner L (1994b) Almost self-optimizing strategies for the adaptive control of diffusion processes. J Optim Theory Appl 81:470–507

Duncan TE, Guo L, Pasik-Duncan B (1999) Adaptive continuous-time linear quadratic Gaussian control. IEEE Trans Autom Control 44:1653–1662

Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York

Goodwin G, Ramadge P, Caines PE (1981) Discrete time stochastic adaptive control. SIAM J Control Optim 19:820–853

Guo L (1995) Convergence and logarithm laws of self-tuning regulators. Automatica 31:435–450

Guo L (1996) Self-convergence of weighted least squares with applications. IEEE Trans Autom Control 41:79–89

Guo L (1997) On critical stability of discrete time adaptive nonlinear control. IEEE Trans Autom Control 42:1488–1499

Guo L, Chen HF (1991) The Astrom-Wittenmark self-tuning regulator revisited and ELS based adaptive trackers. IEEE Trans Autom Control 36:802–812

Kumar PR (1985) A survey of some results in stochastic adaptive control. SIAM J Control Optim 23:329–380

Kumar PR (1990) Convergence of adaptive control schemes with least squares estimates. IEEE Trans Autom Control 35:416–424

Kumar PR, Varaiya P (1986) Stochastic systems, estimation, identification and adaptive control. Prentice-Hall, Englewood Cliffs

Lai TL, Wei CZ (1982) Least square estimation is stochastic regression models with applications to identification and control of dynamic systems. Ann Stat 10:154–166

Ljung L, Soderstrom T (1983) Theory and practice of recursive identification. MIT, Cambridge

Mandl P (1973) On the adaptive control of finite state Markov processes. Z Wahr Verw Geb 27:263–276

Nourian M, Caines PE, Malhame RP (2012) Mean field LQG control in leader-follower stochastic multi-agent systems: likelihood ratio based adaptation. IEEE Trans Autom Control 57:2801–2816

Stochastic Description of Biochemical Networks

João P. Hespanha[1] and Mustafa Khammash[2]
[1]Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA, USA
[2]Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology at Zurich (ETHZ), Basel, Switzerland

Abstract

Conventional deterministic chemical kinetics often breaks down in the small volume of a living cell where cellular species (e.g., genes, mRNAs, etc.) exist in discrete, low copy numbers and react through reaction channels whose timing and order is random. In such an environment, a stochastic chemical kinetics framework that models species abundances as discrete random variables is more suitable. The resulting models consist of continue-time discrete-state Markov chains. Here we describe how such models can be formulated and numerically simulated, and we present some of the key analysis techniques for studying such reactions.

Keywords

Chemical master equation; Gillespie algorithm; Moment dynamics; Stochastic biochemical reactions; Stochastic models

Introduction

The time evolution of a spatially homogeneous mixture of chemically reacting molecules is often modeled using a stochastic formulation, which takes into account the inherent randomness of thermal molecular motion. This formulation is important when modeling complex reactions inside living cells, where small populations of key reactants can set the stage for significant stochastic effects. In this entry, we review the basic stochastic model of chemical reactions and discuss the most common techniques used to simulate and analyze this model.

Stochastic Models of Chemical Reactions

We start by considering a set of N molecular species (reactants) $\mathcal{S}_1, \ldots, \mathcal{S}_N$ that are confined to a fixed volume Ω. These species react through M possible reactions $R_1, \ldots, R_M$. In this formulation of chemical kinetics, we shall assume that the system is in thermal equilibrium and is well mixed. Thus, the reacting molecules move due to their thermal energy. The population of the different reactants is described by a random process $X(t) = (X_1(t) \ldots X_N(t))^T$, where $X_i(t)$ is a random variable that models the abundance (in terms of the number of copies) of molecules of species $\mathcal{S}_i$ in the system at time t. For the allowable reactions, we shall only consider elementary reactions. These could either be monomolecular, $\mathcal{S}_i \rightarrow$ products, or bimolecular, $\mathcal{S}_i + \mathcal{S}_j \rightarrow$ products. Upon the firing of reaction R_k, a transition occurs from some state $X = x_i$ right before the reaction fires to some other state $X = x_i + s_k$, which reflects the change in the population immediately after the reaction has fired. s_k is referred to as the *stoichiometric vector*. The set

Stochastic Description of Biochemical Networks, Table 1 Propensity functions for elementary reactions. The constants c, c', and c'' are related to k, k', and k'', the reaction rate constants from *deterministic* mass-action kinetics. Indeed it can be shown that $c = k$, $c' = k'/\Omega$, and $c'' = 2k''/\Omega$

Reaction type	Propensity function
$S_i \rightarrow$ Products	$c x_i$
$S_i + S_j \rightarrow$ Products $\quad (i \neq j)$	$c' x_i x_j$
$S_i + S_i \rightarrow$ Products	$c'' x_i (x_i - 1)/2$

of allowable M reactions defines the so-called stoichiometry matrix:

$$S = \begin{bmatrix} s_1 \cdots s_M \end{bmatrix}.$$

To each reaction R_k, we associate a *propensity function*, $w_k(x)$ that describes the rate of that reaction. More precisely, $w_k(x)h$ is the probability that, given the system is in state x at time t, R_k fires once in the time interval $[t, t + h)$. The propensity functions for elementary reactions is given in Table 1.

Limiting to the Deterministic Regime

There is an important connection between the stochastic process $X(t)$, as represented by the continuous-time discrete-state Markov chain described above, and the solution of a related deterministic reaction rate equations obtained from mass-action kinetics. To see this, let $\Phi(t) = [\Phi_1(t), \ldots, \Phi_N(t)]^T$ be the vector concentrations of species $S_1, \ldots, S_N$. According to mass-action kinetics, $\Phi(\cdot)$ satisfies the ordinary differential equation:

$$\dot{\Phi} = S f(\Phi(t)), \qquad \Phi(0) = \Phi_0.$$

In order to compare the $\Phi(t)$ with $X(t)$, which represents molecular counts, we divide $X(t)$ by the reaction volume to get $X^\Omega(t) = X(t)/\Omega$. It turns out that $X^\Omega(t)$ *limits* to $\Phi(t)$: According to Kurtz (Ethier and Kurtz 1986), for every $t \geq 0$:

$$\lim_{\Omega \rightarrow \infty} \sup_{s \leq t} \left| X^\Omega(s) - \Phi(s) \right| = 0, \quad \text{almost surely.}$$

S

Hence, over any finite time interval, the stochastic model *converges* to the deterministic mass-action one in the thermodynamic limit. Note that this is only a large volume limit result. In practice, for a fixed volume, a stochastic description may differ considerably from the deterministic description.

Stochastic Simulations

Gillespie's stochastic simulation algorithm (SSA) constructs sample paths for the random process $X(t) = (X_1(t) \ldots . X_N(t))^T$ that are consistent with the stochastic model described above (Gillespie 1976). It consists of the following basic steps:

1. Initialize the state $X(0)$ and set $t = 0$.
2. Draw a random number $\tau \in (0, \infty)$ with exponential distribution and mean equal to $1/ \sum_k w_k(X(t))$.
3. Draw a random number $k \in \{1, 2, \ldots, M\}$ such that the probability of $k = i \in \{1, 2, \ldots, M\}$ is proportional to $w_i(X(t))$.
4. Set $X(t + \tau) = X(t) + s_k$ and $t = t + \tau$.
5. Repeat from (2) until t reaches the desired simulation time.

By running this algorithm multiple times with independent random draws, one can estimate the distribution and statistical moments of the random process $X(t)$.

The Chemical Master Equation (CME)

The *chemical master equation* (CME), also known as the forward Kolmogorov equation, describes the time evolution of the probability that the system is in a given state x. The CME can be derived based on the Markov property of chemical reactions. Suppose the system is in state x at time t. Within an error of order $\mathcal{O}(h^2)$, the following statements apply:

- The probability that an R_k reaction fires exactly once in the time interval $[t, t+h]$ is given by $w_k(x)h$.
- The probability that no reactions fire in the time interval $[t, t + h)$ is given by $1 - \sum_k w_k(x)dx$.

- The probability that more than one reaction fires in the time interval $[t, t + h)$ is zero.

Let $P(x, t)$, denote the probability that the system is in state x at time t. We can express $P(x, t + h)$ as follows:

$$P(x, t + h) = P(x, t) \left(1 - \sum_k w_k(x)h\right)$$
$$+ \sum_k P(x - s_k, t)w_k(x - s_k)h + \mathcal{O}(h^2).$$

The first term on the right-hand side is the probability that the system is already in state x at time t, and no reactions occur in the next h. In the second term on the right-hand side, the kth term in the summation is the probability that the system at time t is an R_k reaction away from being at state x and that an R_k reaction takes place in the next h.

Moving $P(x, t)$ to the left-hand side, dividing by h, and taking the limit as h goes to zero yields the chemical master equation (CME):

$$\frac{dP(x, t)}{dt} = \sum_{k=1}^M \Big(w_k(x - s_k)P(x - s_k, t) - w_k(x)P(x, t)\Big). \quad (1)$$

The CME defines a linear dynamical system in the probabilities of the different states (each state is defined by a specific number of molecules of each of the species). However, there are generally an infinite number of states, and the resulting infinite linear system is not directly solvable. One approach to overcome this difficulty is to approximate the solution of the CME by truncating the states. A particular truncation procedure that gives error bounds is called the finite-state projection (FSP) (Munsky and Khammash 2006). The key idea behind the FSP approach is to keep those states that support the bulk of the probability distribution while projecting the remaining infinite states onto a single "absorbing" state. See Fig. 1.

The left panel in the figure shows the infinite states of a system with two species. The arrows indicate transitions among states caused
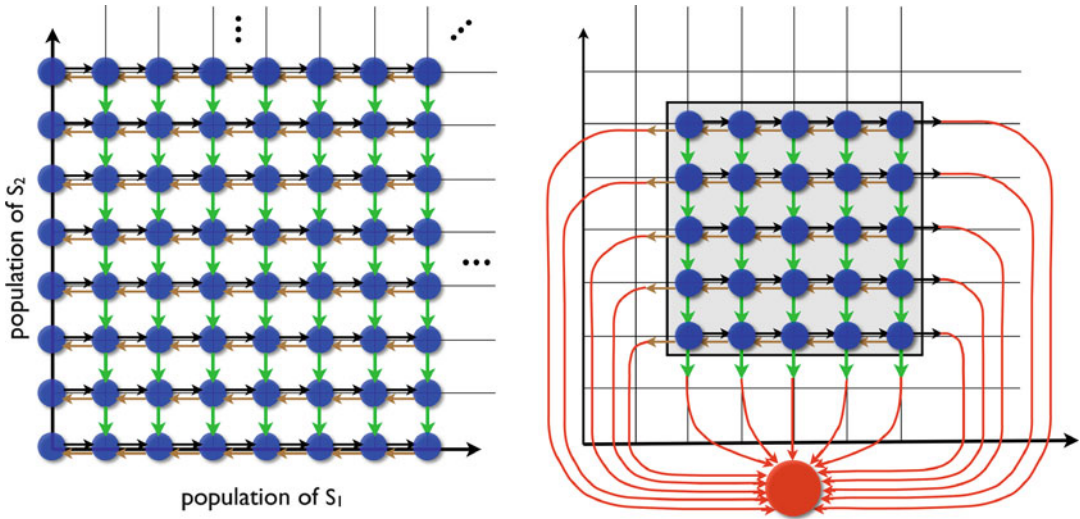
Stochastic Description of Biochemical Networks, Fig. 1 The finite-state projection

by allowable chemical reactions. The underlying stochastic process is a continuous-time discrete-state Markov process. The right panel shows the projected (finite-state) system for a specific projection region (box). The projection is obtained as follows: transitions within the retained sates are kept, while transitions that emanate from these states and end at states outside the box are channeled to a single new absorbing state. Transitions into the box are deleted. The resulting projected system is a finite-state Markov process. The probability of each of its finite states can be computed exactly. It can be shown that the truncation, as defined here, gives a lower bound for the probability for the original full system. The FSP algorithm provides a way for constructing an approximation of the CME that satisfies any prespecified accuracy requirement.

Moment Dynamics

While the probability distribution $P(x, t)$ provides great detail on the state x at time t, often statistical moments of the molecule copy numbers already provide important information about their variability, which motivates the construction

of mathematical models for the evolution of such models over time.

Given a vector of integers $m := (m_1, m_2, \ldots, m_n)$, we use the notation $\mu^{(m)}$ to denote the following uncentered moment of X:

$$\mu^{(m)} := \mathrm{E}[X_1^{m_1} X_2^{m_2} \cdots X_n^{m_n}].$$

Such moment is said to be of order $\sum_i m_i$. With N species, there are exactly N first-order moments $\mathrm{e}[X_i]$, $\forall i \in \{1, 2, \ldots, N\}$, which are just the means; $N(N-1)/2$ second-order moments $\mathrm{e}[X_i^2]$, $\forall i$ and $\mathrm{e}[X_i X_j]$, $\forall i \neq j$, which can be used to compute variances and covariance; $N(N-1)(N-2)/6$ third-order moments; and so on.

Using the CME (1), one can show that

$$\frac{d\mu^{(m)}}{dt} = \mathrm{E}\Big[\sum_k w_k(X) \Big((X_1 + s_{1,k})^{m_1} (X_2 - s_{2,k})^{m_2}$$
$$\cdots (X_N - s_{N,k})^{m_N} - X_1^{m_1} X_2^{m_2} \cdots X_N^{m_N} \Big) \Big],$$

and, because the propensity functions are all polynomials on x (cf. Table 1), the expected value in the right-hand side can actually be written as a linear combination of other uncentered moments of X. This means that if we construct a

vector μ containing all the uncentered moments of x up to some order k, the evolution of μ is determined by a differential equation of the form

$$\frac{d\mu}{dt} = A\mu + B\bar{\mu}, \quad \mu \in \mathbb{R}^K, \ \bar{\mu} \in \mathbb{R}^{\bar{K}} \quad (2)$$

where A and B are appropriately defined matrices and $\bar{\mu}$ is a vector containing moments of order larger than k. The equation (2) is exact, and we call it the *(exact) k-order moment dynamics*, and the integer k is called the *order of truncation*. Note that the dimension K of (2) is always larger than k since there are many moments of each order. In fact, in general, K is of order n^k.

When all chemical reactions have only one reactant, the term $B\bar{\mu}$ does not appear in (2), and we say that the exact moment dynamics are *closed*. However, when at least one chemical reaction has two or more reactants, then the term $B\bar{\mu}$ appears, and we say that the moment dynamics are *open* since (2) depends on the moments in $\bar{\mu}$, which are not part of the state μ. When all chemical reactions are elementary (i.e., with at most two reactants), then all moments in $\bar{\mu}$ are exactly of order $k + 1$.

Moment closure is a procedure by which one approximates the exact (but open) moment dynamics (2) by an approximate (but now closed) equation of the form

$$\dot{v} = Av + B\varphi(v), \quad v \in \mathbb{R}^K \quad (3)$$

where $\varphi(v)$ is a column vector that approximates the moments in $\bar{\mu}$. The function $\varphi(v)$ is called the moment closure function, and (3) is called the *approximate kth-order moment dynamics*. The goal of any moment closure method is to construct $\varphi(v)$ so that the solution v to (3) is close to the solution μ to (2).

There are three main approaches to construct the moment closure function $\varphi(\cdot)$:

1. *Matching-based methods* directly attempt to match the solutions to (2) and (3) (e.g., Singh and Hespanha 2011).
2. *Distribution-based methods* construct $\varphi(\cdot)$ by making reasonable assumptions on the statis-

tical distribution of the molecule counts vector x (e.g., Gomez-Uribe and Verghese 2007).
3. *Large volume methods* construct $\varphi(\cdot)$ by assuming that reactions take place on a large volume (e.g., Van Kampen 2001).

It is important to emphasize that this classification is about methods to *construct* moment closure. It turns out that sometimes different methods lead to the same moment closure function $\varphi(\cdot)$.

Conclusion and Outlook

We have introduced complementary approaches to study the evolution of biochemical networks that exhibit important stochastic effects.

Stochastic simulations permit the construction of sample paths for the molecule counts, which can be averaged to study the ensemble behavior of the system. This type of approach scales well with the number of molecular species, but can be computationally very intensive when the number of reactions is very large. This challenge has led to the development of approximate stochastic simulation algorithms that attempt to simulate multiple reactions in the same simulation step (e.g., Rathinam et al. 2003).

Solving the CME provides the most detailed and accurate approach to characterize the ensemble properties of the molecular counts, but for most biochemical systems such solution cannot be found in closed form, and numerical methods scale exponentially with the number of species. This challenge has led to the development of algorithms that compute approximate solutions to the CME, e.g., by aggregating states with low probability, while keeping track of the error (e.g., Munsky and Khammash 2006).

Moment dynamics is attractive in that the number of kth-order moments only scales polynomially with the number of chemical species, but one only obtains closed dynamics for very simple biochemical networks. This limitation has led to the development of moment closure techniques to approximate the open moment dynamics by a closed system of ordinary differential equations.

Cross-References

Bibliography

Ethier SN, Kurtz TG (1986) Markov processes: characterization and convergence. Wiley series in probability and mathematical statistics: probability and mathematical statistics. Hoboken, New Jersey

Gillespie DT (1976) A general method for numericallysimulating the stochastic time evolution of coupled chemical reactions. J Comput Phys 22:403–434

Gomez-Uribe CA, Verghese GC (2007) Mass fluctuation kinetics: capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. J Chem Phys 126(2):024109–024109–12

Munsky B, Khammash M (2006) The finite state projection algorithm for the solution of the chemical master equation. J Chem Phys 124:044104

Rathinam M, Petzold LR, Cao Y, Gillespie DT (2003) Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. J Chem Phys 119(24):12784–12794

Singh A, Hespanha JP (2011) Approximate moment dynamics for chemically reacting systems. IEEE Trans Autom Control 56(2):414–418

Van Kampen NG (2001) Stochastic processes in physics and chemistry. Elsevier Science, Amsterdam

Stochastic Dynamic Programming

Qing Zhang
Department of Mathematics, The University of Georgia, Athens, GA, USA

Abstract

This article is concerned with one of the traditional approaches for stochastic control problems: Stochastic dynamic programming. Brief descriptions of stochastic dynamic programming methods and related terminology are provided. Two asset-selling examples are presented to illustrate the basic ideas. A list of topics and references are also provided for further reading.

Keywords

Introduction

The term *dynamic programming* was introduced by Richard Bellman in the 1940s. It refers to a method for solving dynamic optimization problems by breaking them down into smaller and simpler subproblems.

To solve a given problem, one often needs to solve each part of the problem (subproblems) and then put together their solutions to obtain an overall solution. Some of these subproblems are of the same type. The idea behind the dynamic programming approach is to solve each subproblem only once in order to reduce the overall computation.

The cornerstone of dynamic programming (DP) is the so-called principle of optimality which is described by Bellman in his 1957 book (Bellman 1957):

> Principle of Optimality: An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

This principle of optimality gives rise to DP (or optimality) equations, which are referred to as Bellman equations in discrete-time optimization problems or Hamilton-Jacobi-Bellman (HJB) equations in continuous-time ones. Such equations provide a necessary condition for optimality in terms of the value of the underlying decision problem. By and large, an optimal control policy in most cases can be obtained by solving the associated Bellman (HJB) equation. In view of this, dynamic programming is a powerful tool for a broad range of control and decision-making problems. When the underlying system is driven by certain type of random

S

disturbance, the corresponding DP approach is referred to as *stochastic dynamic programming*.

Terminology

The following concepts are often used in stochastic dynamic programming.

An **objective function** describes the objective of a given optimization problem (e.g., maximizing profits, minimizing cost, etc.) in terms of the states of the underlying system, decision (control) variables, and possible random disturbance.

State variables represent the information about the current system under consideration. For example, in a manufacturing system, one needs to know the current product inventory in order to decide how much to produce at the moment. In this case, the inventory level would be one of the state variables.

The variables chosen at any time are called the decision or **control variables**. For instance, the rate of production over time in the manufacturing system is a control variable. Typically, control variables are functions of state variables. They affect the future states of the system and the objective function.

In stochastic control problems, the system is also affected by random events (noise). Such noise is referred to system **disturbance**. The noise is often not available a priori. Only their probabilistic distributions are known.

The goal of the optimization problem is to choose control variables over time so as to either maximize or minimize the corresponding objective function. For example, in order to maximize the overall profits, a manufacturing firm has to decide how much to produce over time so as to maximize the revenue by meeting the product demand and minimize the costs associated with inventory. The best possible value of the objective is called **value function**, which is given in terms of the state variables.

In the next two sections, we give two examples to illustrate how stochastic DP methods are used in discrete and continuous time.

An Asset-Selling Example (Discrete Time)

Consider a person wants to sell an asset (e.g., a car or a house). She is offered an amount of money every period (say, a day). Let $v_0, v_1, \ldots, v_{N-1}$ denote the amount of these random offers. Assume they are independent and identically distributed. At the end of each period, the person has to decide whether to accept the offer or reject it. If she accepts the offer, she can put the money in a bank account and receive a fixed interest rate $r > 0$; if she rejects the offer, she waits till the next period. Rejected offers cannot be recycled. In addition, she has to sell her asset by the end of the Nth period and accept the last offer v_{N-1} if all previous offers have been rejected. The goal is to decide when to accept an offer to maximize the overall return at the Nth period.

In this example, for each k, v_k is the random disturbance. The control variables u_k take values in {sell, hold}. The state variables x_k are given by the equations

$$x_0 = 0; \quad x_{k+1} = \begin{cases} \text{sold} & \text{if } u_k = \text{sell} \\ v_k & \text{otherwise.} \end{cases}$$

Let

$$h_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq \text{sold,} \\ 0 & \text{otherwise.} \end{cases}$$

$$h_k(x_k, u_k, v_k) = \begin{cases} (1+r)^{N-k} x_k & \text{if } x_k \neq \text{sold} \\ & \text{and } u_k = \text{sell} \\ 0 & \text{otherwise.} \end{cases}$$

for $k = 0, 1, \ldots, N - 1$.

Then, the payoff function is given by

$$E_{\{v_k\}} \left(h_N(x_N) + \sum_{k=0}^{N-1} h_k(x_k, u_k, v_k) \right).$$

Here, $E_{\{v_k\}}$ represents the expected value over $\{v_k\}$. The corresponding value functions $V_k(x_k)$ satisfy the following Bellman equations:

$$V_N(x_N) = \begin{cases} x_N & \text{if } x_N \neq \text{sold,} \\ 0 & \text{otherwise.} \end{cases}$$

$$V_k(x_k) = \begin{cases} \max\left((1+r)^{N-k} x_k, E V_{k+1}(v_k)\right) & \text{if } x_k \neq \text{sold} \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{for } k = 0, 1, \ldots, N-1.$$

The optimal selling rule can be given as (assuming $x_k \neq$ sold) (see Bertsekas 1987):

accept the offer

$$v_{k-1} = x_k \text{ if } (1+r)^{N-k} x_k \geq E V_{k+1}(v_k),$$

reject the offer

$$v_{k-1} = x_k \text{ if } (1+r)^{N-k} x_k < E V_{k+1}(v_k).$$

Given the distribution for v_k, one can compute V_k backwards and solve the Bellman equations, which in turn leads to the above optimal selling rule.

Note that such backward iteration only works with finite horizon dynamic programming. When working with an infinite horizon (discounted or long-run average) payoff function, often used methods are value iteration (successive approximation) and policy iteration. The idea is to construct a sequence of functions recursively so that they converge pointwise to the value function. For description of these iteration methods, their convergence properties, and error bound analysis, we refer the reader to Bertsekas (1987).

Next, we consider a continuous-time asset-selling problem.

An Asset-Selling Example (Continuous Time)

Suppose a person wants to sell her asset. The price x_t at time $t \in [0, \infty)$ of her asset is given by a stochastic differential equation

$$\frac{dx_t}{x_t} = \mu dt + \sigma dw_t,$$

where μ and σ are known constants and w_t is the standard Brownian motion representing the disturbance. Suppose the transaction cost is K and the discount rate r. She has to decide when to sell her asset to maximize an expected return. In this example, the state variable is price x_t, control variable is a function of selling time τ, and the payoff function is given by

$$J(x, \tau) = E e^{-r\tau} (x_\tau - K).$$

Let $V(x)$ denote the value function, i.e., $V(x) = \sup_\tau J(x, \tau)$. Then the associate HJB equation is given by

$$\min\left\{ r V(x) - x\mu \frac{dV(x)}{dx} - \frac{x^2 \sigma^2}{2} \frac{d^2 V(x)}{dx^2}, \right.$$
$$\left. V(x) - K \right\} = 0. \qquad (1)$$

Let

$$x^* = \frac{K\beta}{\beta - 1},$$

where

$$\beta = \frac{1}{\sigma^2} \left(\frac{\sigma^2}{2} - \mu + \sqrt{\left(\mu - \frac{\sigma^2}{2}\right)^2 + 2r\sigma^2} \right).$$

Then the optimal selling rule can be given as (see Øksendal 2007):

$$\begin{cases} \text{sell} & \text{if } x_t \geq x^*, \\ \text{hold} & \text{if } x_t < x^*. \end{cases}$$

In general, to solve an optimal control problem via the DP approach, one first needs to solve the associate Bellman (HJB) equations. Then, these

S

solutions can be used to come up with an optimal control policy. For example, in the above case, given the value function $V(x)$, one should hold if

$$rV(x) - x\mu \frac{dV(x)}{dx} - \frac{x^2\sigma^2}{2}\frac{d^2V(x)}{dx^2} = 0$$

and sell when $V(x) - K = 0$. The threshold level x^* is the exact dividing point between the first part equals zero and the second part vanishes. In addition, one can also provide a theoretical justification in terms of a verification theorem to show that the solution obtained this way is indeed optimal (see Fleming and Rishel (1975), Fleming and Soner (2006), or Yong and Zhou (1999)).

HJB Equation Characterization and Computational Methods

In continuous-time optimal control problem, one major difficulty that arises in solving the associated HJB equations (e.g., (1)) is the characterization of the solutions. In most cases, there is no guarantee that the derivatives or partial derivatives exist. In this connection, the concept of viscosity solutions developed by Crandall and Lions in the 1980s can often be used to characterize the solutions and their uniqueness. We refer the reader to Fleming and Soner (2006) for related literature and applications. In addition, we would like to point out that closed-form solutions are rare in stochastic control theory and difficult to obtain in most cases. In many applications, one needs to resort to computational methods. One typical way to solve an HJB equation is the finite difference methods. An alternative is Kushner's Markov chain approximation methods; see Kushner and Dupuis (1992).

Summary and Future Directions

In this article, we have briefly stated stochastic DP methods, showed how they work in two simple examples, and discussed related issues. One serious limitation of the DP approach is the so-called curse of dimensionality. In other words, the DP does not work for problems with high dimensionality. Various efforts have been devoted to search for approximate solutions. One approach developed in recent years is the multi-time-scale approach. The idea is to classify random events according to the frequency of their occurrence. Frequent occurring events are grouped together and treated as a single "state" to achieve the reduction of dimensionality. We refer the reader to Yin and Zhang (2005, 2013) for related literature and theoretical development. Finally, we would like to mention that stochastic DP has been used in many applications in economics, engineering, management science, and finance. Some applications can be found in Sethi and Thompson (2000). Additional references are also provided at the end for further reading.

Cross-References

▶ Backward Stochastic Differential Equations and Related Control Problems
▶ Numerical Methods for Continuous-Time Stochastic Control Problems
▶ Risk-Sensitive Stochastic Control
▶ Stochastic Adaptive Control
▶ Stochastic Linear-Quadratic Control
▶ Stochastic Maximum Principle

Bibliography

Bellman RE (1957) Dynamic programming. Princeton University Press, Princeton

Bertsekas DP (1987) Dynamic programming. Prentice Hall, Englewood Cliffs

Davis MHA (1993) Markov models and optimization. Chapman & Hall, London

Elliott RJ, Aggoun L, Moore JB (1995) Hidden Markov models: estimation and control. Springer, New York

Fleming WH, Rishel RW (1975) Deterministic and stochastic optimal control. Springer, New York

Fleming WH, Soner HM (2006) Controlled Markov processes and viscosity solutions, 2nd edn. Springer, New York

Hernandez-Lerma O, Lasserre JB (1996) Discrete-time Markov control processes: basic optimality criteria. Springer, New York

Kushner HJ, Dupuis PG (1992) Numerical methods for stochastic control problems in continuous time. Springer, New York

Kushner HJ, Yin G (1997) Stochastic approximation algorithms and applications. Springer, New York

Øksendal B (2007) Stochastic differential equations, 6th edn. Springer, New York

Pham H (2009) Continuous-time stochastic control and optimization with financial applications. Springer, New York

Sethi SP, Thompson GL (2000) Optimal control theory: applications to management science and economics, 2nd edn. Kluwer, Boston

Sethi SP, Zhang Q (1994) Hierarchical decision making in stochastic manufacturing systems. Birkhäuser, Boston

Yin G, Zhang Q (2005) Discrete-time Markov chains: two-time-scale methods and applications. Springer, New York

Yin G, Zhang Q (2013) Continuous-time Markov chains and applications: a two-time-scale approach, 2nd edn. Springer, New York

Yin G, Zhu C (2010) Hybrid switching diffusions: properties and applications. Springer, New York

Yong J, Zhou XY (1999) Stochastic control: Hamiltonian systems and HJB equations. Springer, New York

Stochastic Games and Learning

Krzysztof Szajowski
Faculty of Fundamental Problems of
Technology, Institute of Mathematics and
Computer Science, Wroclaw University of
Technology, Wroclaw, Poland

Abstract

A stochastic game was introduced by Lloyd Shapley in the early 1950s. It is a dynamic game with *probabilistic transitions* played by one or more players. The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain *state*. The players select actions, and each player receives a *payoff* that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the players. The procedure is repeated at the new state, and the play continues for a finite or infinite number of stages. The total payoff to a player is often taken to be the discounted sum of the stage payoffs
or the limit inferior of the averages of the stage payoffs.

A learning problem arises when the agent does not know the reward function or the state transition probabilities. If an agent directly learns about its optimal policy without knowing either the reward function or the state transition function, such an approach is called *model-free reinforcement learning*. Q-learning is an example of such a model.

Q-learning has been extended to a noncooperative multi-agent context, using the framework of general-sum stochastic games. A learning agent maintains Q-functions over joint actions and performs updates based on assuming Nash equilibrium behavior over the current Q-values. The challenge is convergence of the learning protocol.

Keywords

Introduction

A Stochastic Game

Definition 1 (Stochastic games) A stochastic game is a dynamic game with probabilistic transitions played by one or more players. The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain state. The players select *actions*, and each player receives *a payoff* that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the players. The process is repeated at the new state, and the play continues for a finite or infinite number of stages.

The total payoff to a player can be defined in various ways. It depends on the payoffs at each stage and strategies chosen by players. The aim of the players is to control their total payoffs in the game by appropriate actions.

S

The notion of a stochastic game was introduced by Lloyd Shapley (1953) in the early 1950s. Stochastic games generalize both Markov decision processes (see also MDP) and repeated games. A repeated game is equivalent to a stochastic game with a single state. The stochastic game is played in discrete time with past history as common knowledge for all the players. An *individual strategy* for a player is a map which associates with each given history a probability distribution on the set of actions available to the players. The players' actions at stage n determines the players' payoffs at this stage and the state $s \in \mathfrak{S}$ at stage $n + 1$.

Learning

Learning is acquiring new, or modifying and reinforcing existing, knowledge, behaviors, skills, values, or preferences, and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals, and some machines which will be later called *agents*. In the context of this entry, learning refers to a particular class of stochastic game theoretical models.

Definition 2 (Learning in stochastic games) A learning problem arises when an agent does not know the reward function or the state transition probabilities. If the agent directly learns about its optimal policy without knowing either the reward function or the state transition function, such an approach is called *model-free reinforcement learning*. Q-learning is an example of such a model.

Learning models constitute a branch of larger literature. Players follow a form of behavioral rule, such as imitation, regret minimization, or reinforcement. Learning models are most appropriate in settings where players have a good understanding of their strategic environment and where the stakes are high enough to make forecasting and optimization worthwhile. The known approaches are formulated as *minimax-Q* (Littman 1994), Nash-Q (Hu and Wellman 1998), tinkering with learning rates ("Win or Learn Fast"-WoLF, Bowling and Veloso 2001) and multiple timescale Q-learning (Leslie and Collins 2005).

Model of Stochastic Game

Let us assume that the environment is modeled by the probability space $(\Omega, \mathcal{F}, \mathbf{P})$. An N-*person stochastic game* is described by the objects $(\mathfrak{N}, \mathfrak{S}, X_k, A_k, r_k, q)$ with the interpretation that:

1. $\mathfrak{N}$ is a set of players, with $|\mathfrak{N}| = N \in \mathbb{N}$.
2. $\mathfrak{S}$ is the *set of states* of the game, and it is finite.
3. $\overrightarrow{X} = X_1 \times X_2 \times \ldots \times X_N$ is the *state of actions*, where X_k is a nonempty, finite space of actions for player k.
4. A_k's are correspondences from $\mathfrak{S}$ into nonempty subsets of X_k. For each $s \in \mathfrak{S}$, $A_k(s)$ represents the *set of actions* available to player k in state s. For $s \in \mathfrak{S}$, denote $\overrightarrow{A}(s) = A_1(s) \times A_2(s) \times \ldots \times A_N(s)$.
5. $r_k : \mathfrak{S} \times \overrightarrow{X} \to \mathfrak{R}$ is a payoff function for player k.
6. q is a transition probability from $\mathfrak{S} \times \overrightarrow{X}$ to $\mathfrak{S}$, called the *law of motion* among states. If s is a state at a certain stage of the game and the players select $\overrightarrow{x} \in \overrightarrow{A}(s)$, then $q\left(\cdot | s, \overrightarrow{x}\right)$ is the probability distribution of the next state of the game.

The stochastic game generates two processes:
1. $\{\sigma_n\}_{n=1}^{T}$ with values in $\mathfrak{S}$
2. $\{\alpha_n\}_{n=1}^{T}$ with values in $\overrightarrow{X}$

Strategies

Let $\mathfrak{H} = \mathfrak{S}_1 \times \overrightarrow{X}_1 \times \mathfrak{S}_2 \times \cdots$ be the space of all infinite histories of the game and $\mathfrak{H}_n = \mathfrak{S}_1 \times \overrightarrow{X}_1 \times \mathfrak{S}_2 \times \overrightarrow{X}_2 \times \cdots \mathfrak{S}_n$ the histories up to stage n.

Definition 3 A player's *strategy* $\pi = \{\alpha_n\}_{n=1}^{T}$ consists of random maps $\alpha_n : \Omega \times \mathfrak{H}_n \to X$. In other words, the strategy associates with each given history a probability distribution dependent on the set of actions available to the player. If α_n is dependent on the history only, it is called deterministic.

The mathematical description of the strategies can be made as follows:
1. For player $i \in \mathbb{N}$, a deterministic strategy specifies a choice of actions for the player at every stage of every possible history.

2. A mixed strategy is a probability distribution over deterministic strategies.
3. Restricted classes of strategies:
   1. A behavioral strategy – a mixed strategy in which the mixing takes place at each history independently.
   2. A Markov strategy – a behavioral strategy such that for each time t, the distribution over actions depends only on the current state, but the distribution may be different at time t than at time $t' \neq t$.
   3. A stationary strategy – a Markov strategy in which the distribution over actions depends only on the current state (not on the time t).

The Total Payoff Types

For any profile of strategies $\pi = (\pi_1, \ldots, \pi_N)$ of the players and every initial state $s_1 = s \in \mathfrak{S}$, a probability measure P_s^π and a stochastic process $\{\sigma_n, \alpha_n\}$ are defined on $\mathfrak{H}$ in a canonical way, where the random variables σ_n and α_n describe the state and the actions chosen by the players, respectively, on the nth stage of the game. Let us define E_s^π the expectation operator with respect to the probability measure P_s^π. For each profile of strategies $\pi = (\pi_1, \ldots, \pi_N)$ and every initial state $s \in \mathfrak{S}$, the following are considered:

1. The *expected T-stage payoff* to player k, for any finite horizon T, defined as

$$\Phi_k^T(\pi)(s) = E_s^\pi \left(\sum_{n=1}^{T} r_k(\sigma_n, \alpha_n) \right)$$

2. The β-discounted expected payoff to player k, where $\beta \in (0, 1)$ is called the *discount factor*, defined as

$$\Phi_k^\beta(\pi)(s) = E_s^\pi \left(\sum_{n=1}^{\infty} \beta^{n-1} r_k(\sigma_n, \alpha_n) \right)$$

3. The *average payoff per unit time* for player k defined as

$$\Phi_k(\pi)(s) = \lim_T \sup \frac{1}{T} \Phi_k^T(\pi)(s)$$

Equilibria

Let $\pi^* = (\pi_1^*, \ldots, \pi_N^*) \in \Pi$ be a fixed profile of the players' strategies. For any strategy $\pi_k \in \Pi_k$ of player k, we write (π_{-k}^*, π_k) to denote the strategy profile obtained from π^* by replacing π_k^* with π_k.

Definition 4 (A Nash equilibrium) A strategy profile $\pi^* = (\pi_1^*, \ldots, \pi_N^*) \in \Pi$ is called a *Nash equilibrium* (in Π) for the average payoff stochastic game if no unilateral deviations from it are profitable, that is, for each $s \in S$,

$$\Phi_k(\pi^*)(s) \geq \Phi_k(\pi_{-k}^*, \pi_k)(s)$$

for every player k and any strategy π_k.

Definition 5 (An ε-Nash equilibrium) A strategy profile $\pi^* = (\pi_1^*, \ldots, \pi_N^*)$ is called an ε-*(Nash) equilibrium* of the average payoff stochastic game if for every $k \in \mathfrak{N}$, we have

$$\Phi_k(\pi^*)(s) \geq \Phi_k(\pi_{-k}^*, \pi_k)(s) - \epsilon,$$

for the given $\varepsilon > 0$ and all π_k.

Nash equilibria and ε-Nash equilibria are analogously defined for the T-stage stochastic games, β-discounted stochastic games, and the average payoff per unit time stochastic games.

Construction of an Equilibrium

For stochastic games with a finite state space and finite action spaces, the existence of a stationary equilibrium has been shown (cf. Herings and Peeters 2004). The stationary strategies at time t do not depend on the entire history of the game up to that time. This allows reduction of the problem of finding discounted stationary equilibria in a general n-person stochastic game to that of finding a global minimum in a nonlinear program with linear constraints. Solving this nonlinear program is equivalent to solving a certain nonlinear system for which it is known that the objective value in the global minimum is zero (cf. Filar et al. 1991). However, as is noted by Breton (1991), the convergence of an optimization algorithm to the global optimum is not guaranteed.

S

The solution of the finite horizon finite stochastic game can be construct by dynamic programming (see, e.g., Nowak and Szajowski 1998; Tijms 2012). For discounted games, the solution construction is based on an equivalence (the two-person case is presented here for simplicity):

1. $\left(\pi_1^*, \pi_2^*\right)$ is an equilibrium point in the discounted stochastic game with equilibrium payoffs $\left(\Phi_1^\beta\left(\overrightarrow{\pi}^*\right), \Phi_2^\beta\left(\overrightarrow{\pi}^*\right)\right)$.

2. For each $s \in \mathfrak{S}$, the pair $\left(\pi_1^*(s), \pi_2^*(s)\right)$ constitutes an equilibrium point in the static bimatrix game $(B_1(s), B_2(s))$ with equilibrium payoffs $\left(\Phi_1^\beta\left(s, \overrightarrow{\pi}^*\right), \Phi_2^\beta\left(s, \overrightarrow{\pi}^*\right)\right)$, where for players $k = 1, 2$, and pure actions $(a_1, a_2) \in A_1(s) \times A_2(s)$, an admissible action space at state s, the elements of $B_k(s)$ related to (a_1, a_2)

$$
\begin{aligned}
b_k(s, a_1, a_2) := &(1 - \beta) r_k(s, a_1, a_2) \\
&+ \beta E_s^{(a_1, a_2)} \Phi_k^\beta\left(\overrightarrow{\pi}^*\right)
\end{aligned} \quad (1)
$$

An algorithm for recursive computation of stationary equilibria in stochastic games can be derived from (1). It starts with bimatrix games with $\beta = 0$, and then a careful equilibrium selection process guarantees its convergence under mild assumptions on the model (see, e.g., Herings and Peeters 2004).

A Brief History of the Research on Stochastic Games

The notion of a stochastic game was introduced by Shapley (1953) in the early 1950s. It is a dynamic game with *probabilistic transitions* played by one or more players. The game is played in a sequence of stages. At the beginning of each stage, the game is in a certain *state*. The players select actions, and each player receives a *payoff* that depends on the current state and the chosen actions. The game then moves to a new random state whose distribution depends on the previous state and the actions chosen by the players. The process is repeated at the new state, and the play continues for a finite or an infinite number of stages. The total payoff to a player is often taken to be the discounted sum of the stage

payoffs or the limit inferior of the averages of the stage payoffs.

The theory of nonzero-sum stochastic games with the average payoffs per unit time for the players started with the papers by Rogers (1969) and Sobel (1971). They considered finite state spaces only and assumed that the transition probability matrices induced by any stationary strategies of the players are irreducible. Until now, only special classes of nonzero-sum average payoff stochastic games have been shown to possess Nash equilibria (or ε-equilibria). A review of various cases and results for generalization to infinite state spaces can be found in the survey paper by Nowak and Szajowski (1998).

Learning in Stochastic Game

The problem of an agent learning to act in an unknown world is both challenging and interesting. Reinforcement learning has been successful at finding optimal control policies for a single agent operating in a stationary environment, specifically a Markov decision process. Learning to act in multi-agent systems offers additional challenges (see the following surveys: Shoham and Leyton-Brown 2009, Chap. 7; Weiß and Sen 1996; Buşoniu et al. 2010). We provide here, an overview of a general idea of learning for single and multi-agent systems:

1. Goals of single-agent reinforcement learning are to determine the optimal value and a control policy which maximizes the payoff. The model of such a system can be built based on the framework of Markov decision processes with discounted payoff. Suppose the policy is stationary and defined by a function $h : \mathfrak{S} \to X$. Such a policy defines what action should be taken in each state: $\alpha_n(\cdot) := h(\cdot)$. There are various ways to learn the optimal policy. The most straightforward way is based on the Q-values: $Q^h(s, a) = \sum_{j=0}^{\infty} \beta_{j+1}^{jr}$. The greedy action is $a = \arg\max_{a' \in A(s)} Q^h(s, a')$ (see the article on Q-learning in Reinforcement learning).

2. Multi-agent reinforcement learning can be employed to solve a single task, or an agent may be required to perform a task in an environment with other agents, either human, robot, or software ones. In either case, from an agent's perspective, the world is not stationary. In particular, the behavior of the other agents may change as they also learn to better perform their tasks. This type of a multi-agent nonstationary world creates a difficult problem for learning to act in these environments. Such a nonstationary scenario can be viewed as a game with multiple players. In game theory, in the study of such problems, there is generally an underlying assumption that the players have similar adaptation and learning abilities. Therefore, the actions of each agent affect the task achievement of the other agents. It allows to build the value of the game and an equilibrium strategy profile in following steps.

Stochastic games can be seen as an extension of the single-agent Markov decision process framework to include multiple agents whose actions all impact the resulting rewards and the next state. They can also be viewed as an extension of the framework of matrix games. Such a view emphasizes the difficulty of finding the optimal behavior in stochastic games since the optimal behavior of any one agent depends on the behavior of other agents. A comprehensive study of the multi-agent learning techniques for stochastic games does not yet exist. For the interested reader, there are monographs by Fudenberg and Levine (1998) and Shoham and Leyton-Brown (2009) and the special issue of the journal *Artificial Intelligence* (Vohra and Wellman 2007), which could be consulted.

Despite its interesting properties, Q-learning is a very slow method that requires a long period of training for learning an acceptable policy. In practice, to reduce the problem, there are parallel computing implementation models of Q-learning.

Summary and Future Directions

Details concerning solution concepts for stochastic games can be found in Filar and Vrieze (1997).

The refinements of the Nash equilibrium concept have been known in the economic dynamic games (see Myerson 1978). The Nash equilibrium concept may be extended gradually when the rules of the game are interpreted in a broader sense, so as to allow preplay or even intraplay communication. A well-known extension of the Nash equilibrium is Aumann's correlated equilibrium (see Aumann 1987), which depends only on the normal form of the game. Two other solution concepts for multistage games have been proposed by Forges (1986): the extensive form correlated equilibrium, where the players can observe private exogenous signals at every stage, and the communication equilibrium, where the players are furthermore allowed to transmit inputs to an appropriate device at every stage. An application of the notion of correlated equilibria for stochastic games can be found in Nowak and Szajowski (1998).

In economics, in the context of economic growth problems, Ramsey (1928) has introduced an *overtaking optimality* and independently (Rubinstein 1979) for repeated games. The criterion has been investigated for some stochastic games by Carlson and Haurie (1995) and Nowak (2008), and others. The existence of overtaking optimal strategies is a subtle issue, and there are counterexamples showing that one has to be careful with making statements on overtaking optimality.

Regarding a stochastic game and learning, let us mention that the first idea can be found in the papers by Brown (1951) and Robinson (1951). Some convergence results for a fictitious play have been given by Shoham and Leyton-Brown (2009) in Theorem 7.2.5. An important example showing non-convergence was given by Shapley (1964). In multi-person stochastic games and learning, convergence to equilibria is a basic stability requirement (see, e.g., Greenwald and Hall 2003; Hu and Wellman 2003). This means that the agents' strategies should eventually converge to a coordinated equilibrium. Nash equilibrium is most frequently used, but their usefulness is suspected. For instance, in Shoham and Leyton-Brown (2009), there is an argument that the link between stage-wise convergence to

S

Nash equilibria and the performance in stochastic games is unclear.

Cross-References

- ▶ Dynamic Noncooperative games
- ▶ Evolutionary Games
- ▶ Iterative Learning Control
- ▶ Learning in Games
- ▶ Stochastic Adaptive control

Bibliography

Aumann RJ (1987) Correlated equilibrium as an expression of Bayesian rationality. Econometrica 55:1–18. doi:10.2307/1911154

Bowling M, Veloso M (2001) Rational and convergent learning in stochastic games. In: Proceedings of the 17th international joint conference on artificial intelligence (IJCAI), Seattle, pp 1021–1026

Breton M (1991) Algorithms for stochastic games. In: Raghavan TES, Ferguson TS, Parthasarathy T, Vrieze OJ (eds) Stochastic games and related topics: in honor of Professor L. S. Shapley, vol 7. Springer Netherlands, Dordrecht, pp 45–57. doi:10.1007/978-94-011-3760-7_5

Brown GW (1951) Iterative solution of games by fictitious play. In: Koopmans TC (ed) Activity analysis of production and allocation. Wiley, New York, Chap. XXIV, pp 374–376

Buşoniu L, Babuška R, Schutter BD (2010) Multi-agent reinforcement learning: an overview. In: Srinivasan D, Jain LC (eds) Innovations in multi-agent systems and application–1. Springer, Berlin, pp 183–221

Carlson D, Haurie A (1995) A turnpike theory for infinite horizon open-loop differential games with decoupled controls. In: Olsder GJ (ed) New trends in dynamic games and applications. Annals of the international society of dynamic games, vol 3. Birkhäuser, Boston, pp 353–376

Filar J, Vrieze K (1997) Competitive Markov decision processes. Springer, New York

Filar JA, Schultz TA, Thuijsman F, Vrieze OJ (1991) Nonlinear programming and stationary equilibria in stochastic games. Math Program 50(2, Ser A):227–237. doi:10.1007/BF01594936

Forges F (1986) An approach to communication equilibria. Econometrica 54:1375–1385. doi:10.2307/1914304

Fudenberg D, Levine DK (1998) The theory of learning in games, vol 2. MIT, Cambridge

Greenwald A, Hall K (2003) Correlated-Q learning. In: Proceedings 20th international conference on machine learning (ISML-03), Washington, DC, 21–24 Aug 2003, pp 242–249

Herings PJ-J, Peeters RJAP (2004) Stationary equilibria in stochastic games: structure, selection, and computation. J Econ Theory 118(1):32–60. doi:10.1016/j.jet.2003.10.001

Hu J, Wellman MP (1998) Multiagent reinforcement learning: theoretical framework and an algorithm. In: Proceedings of the 15th international conference on machine learning, New Brunswick, pp 242–250

Hu J, Wellman MP (2003) Nash Q-learning for general-sum stochastic games. J Mach Learn Res 4:1039–1069

Leslie DS, Collins EJ (2005) Individual Q-learning in normal form games. SIAM J Control Optim 44(2):495–514. doi:10.1137/S0363012903437976

Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 13th international conference on machine learning, New Brunswick, pp 157–163

Myerson RB (1978) Refinements of the Nash equilibrium concept. Int J Game Theory 7(2):73–80. doi:10.1007/BF01753236

Nowak AS (2008) Equilibrium in a dynamic game of capital accumulation with the overtaking criterion. Econ Lett 99(2):233–237. doi:10.1016/j.econlet.2007.05.033

Nowak AS, Szajowski K (1998) Nonzerosum stochastic games. In: Bardi M, Raghavan TES, Parthasarathy T (eds) Stochastic and differential games: theory and numerical methods. Annals of the international society of dynamic games, vol 4. Birkhäser, Boston, pp 297–342. doi:10.1007/978-1-4612-1592-9_7

Ramsey F (1928) A mathematical theory of savings. Econ J 38:543–559

Robinson J (1951) An iterative method of solving a game. Ann Math 2(54):296–301. doi:10.2307/1969530

Rogers PD (1969) Nonzero-sum stochastic games, PhD thesis, University of California, Berkeley. ProQuest LLC, Ann Arbor

Rubinstein A (1979) Equilibrium in supergames with the overtaking criterion. J Econ Theory 21:1–9. doi:10.1016/0022-0531(79)90002-4

Shapley L (1953) Stochastic games. Proc Natl Acad Sci USA 39:1095–1100. doi:10.1073/pnas.39.10.1095

Shapley L (1964) Some topics in two-person games. Ann Math Stud 52:1–28

Shoham Y, Leyton-Brown K (2009) Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511811654

Sobel MJ (1971) Noncooperative stochastic games. Ann Math Stat 42:1930–1935. doi:10.1214/aoms/1177693059

Tijms H (2012) Stochastic games and dynamic programming. Asia Pac Math Newsl 2(3):6–10

Vohra R, Wellman M (eds) (2007) Foundations of multi-agent learning. Artif Intell 171:363–452

Weiß G, Sen S (eds) (1996) Adaption and learning in multi-agent Systems. In: Proceedings of the IJCAI'95 workshop, Montréal, 21 Aug 1995, vol 1042. Springer, Berlin. doi:10.1007/3-540-60923-7

Stochastic Linear-Quadratic Control

Shanjian Tang
Fudan University, Shanghai, China

Abstract

In this short article, we briefly review some major historical studies and recent progress on continuous-time stochastic linear-quadratic (SLQ) control and related mean-variance (MV) hedging.

Keywords

Bellman's quasilinearization; BMO-martingale; Mean-variance hedging; Monotone convergence; Quadratic backward stochastic differential equations; Riccati equation

Introduction

A stochastic linear-quadratic (SLQ) control problem is the optimal control of a linear stochastic dynamic equation subject to an expected quadratic cost functional of the system state and control. As shown in Athans (1971), it is a typical case of optimal stochastic control both in theory and application. Due to the linearity of the system dynamics and the quadratic feature of the cost functions, the optimal control law is usually synthesized into a feedback (also called closed) form of the optimal state, and the corresponding proportional coefficients are specified by the associated Riccati equation. In what follows, we restrict our exposition within the continuous-time SLQ problem, and further, mainly for the finite-horizon case.

The initial study on the continuous-time SLQ problem seems to be due to Florentin (1961). However, his linear stochastic control system is assumed to be Gaussian. That is, the system noise is additive and has neither multiplication with the state nor with the control. Such a case is usually

termed as the linear-quadratic Gaussian (LQG) problem, and in the case of complete observation, the optimal feedback law remains to be invariant when the white noise vanishes. The continuous-time partially observable case was first discussed by Potter (1964) and a more general formulation was later given by Wonham (1968a). It is proved that the optimal control can be obtained by the following two separate steps: (1) generate the conditional mean estimate of the current state using a Kalman filter and (2) optimally feed back as if the conditional mean state estimate was the true state of the system. This result is referred to as the certainty equivalence principle or the strict separation theorem. Different assumptions were discussed by Tse (1971) for the separation of control and state estimation.

Wonham (1967, 1968b, 1970) investigated the SLQ problem in a fairly general systematic framework. In the first two papers, his stochastic system is able to admit a state-dependent noise. Finally, Wonham (1970) considered the following very general (admitting both state- and control-dependent noise) linear stochastic differential system driven by a d-dimensional Brownian motion $W = (W^1, W^2, \cdots, W^d)$:

$$
X_t = x + \int_0^t (A_s X_s + B_s u_s)\, dt
$$
$$
+ \int_0^t \sum_{i=1}^d (C_s^i X_s + D_s^i u_s)\, dW_s^i, \;\; t \in [0, T];
$$

and the following cost functional:

$$
J(u) = E\langle M X_T, X_T \rangle
$$
$$
+ E \int_0^T [\langle Q_t X_t, X_t \rangle + \langle N_t u_t, u_t \rangle]\, dt.
$$

Here, $T > 0$, $X_t \in R^n$ is the state at time t, and $u_t \in R^m$ is the control at time t. Assume that all the coefficients $A, B; C^i, D^i, i = 1, 2, \ldots, d; Q, N$ are piecewisely continuous matrix-valued (of suitable dimensions) functions of time, and M, Q_t are nonnegative matrices and N_t is uniformly positive. Wonham (1970) gave the following Riccati equation:

S

$$\begin{cases} -\dot{K}_t = A_t^* K_t + K_t A_t + C_t^{i*} K_t C_t^i - \Gamma_t(K_t)(N_t + D_t^{i*} K_t D_t^i)\Gamma_t(K_t), & t \in [0, T); \\ K_T = M. \end{cases} \quad (1)$$

Here, the asterisk stands for transpose, the repeated superscripts imply summation from 1 to d, and the function Γ is defined by

$$\Gamma_t(K) := -(N_t + D_t^i K D_t^i)^{-1}(KB_t + C_t^{i*} K D_t^i)^*$$

for time $t \in [0, T]$ and any $K \in \mathscr{S}_+^n :=$ {all nonnegative $n \times n$ matrices}. This Riccati equation is a nonlinear ordinary differential equation (ODE). Since the nonlinear term $\Gamma_t(K)(N_t + D_t^{i*} K D_t^i)\Gamma_t(K)$ in the right-hand side is not uniformly Lipschitz in K in general, the standard existence and uniqueness theorem of ODEs does not directly tell whether this Riccati equation has a unique continuous solution in $\mathscr{S}_+^n$. To solve this issue, Wonham (1970) used Bellman's principle of quasilinearization and constructed the following sequence of successive linear approximating matrix-valued ODEs.

Define for $(t, K, \tilde{\Gamma}) \in [0, T] \times R^{n \times n} \times R^{m \times n}$,

$$\begin{aligned} F_t(K, \tilde{\Gamma}) := &[A_t + B_t \tilde{\Gamma}]^* K + K[A_t + B_t \tilde{\Gamma}] \\ &+ [C_t^i + D_t^i \tilde{\Gamma}]^* K[C_t^i + D_t^i \tilde{\Gamma}] \\ &+ Q_t + \tilde{\Gamma}^* N_t \tilde{\Gamma}. \end{aligned} \quad (2)$$

For $K \in \mathscr{S}_+^n$, the matrix $F_t(K, \tilde{\Gamma}) - F_t(K, \Gamma_t(K))$ is nonnegative, that is,

$$F_t(K, \tilde{\Gamma}) \geq F_t(K, \Gamma_t(K)), \quad \forall \tilde{\Gamma} \in R^{m \times n}. \quad (3)$$

Riccati equation (1) can then be written into the following form:

$$\begin{cases} -\dot{K}_t = F_t(K_t, \Gamma_t(K_t)), & t \in [0, T); \\ K_T = M. \end{cases} \quad (4)$$

The iterating linear approximations are therefore structured as follows: Set $K^0 \equiv M$ and for $l = 1, 2, \ldots,$

$$\begin{cases} -\dot{K}_t^l = F_t(K_t^l, \Gamma_t(K_t^{l-1})), & t \in [0, T); \\ K_T^l = M. \end{cases} \quad (5)$$

Using the above minimal property (3) of $F_t(K, \cdot)$ at $\Gamma_t(K)$, Wonham showed that the unique nonnegative solution K^l of ODE (5) is monotonically decreasing in the sequential number $l = 1, 2, \ldots$. Using the method of monotone convergence, the sequence of solutions $\{K^l\}$ is shown to converge to some $K \in \mathscr{S}_+^n$, which turns out to solve Riccati equation (1).

The Case of Random Coefficients and Backward Stochastic Riccati Equation

Bismut (1976, 1978) are the first studies on the SLQ problem with random coefficients. Let $\{\mathscr{F}_t, t \in [0, T]\}$ be the completed natural filtration of W. When the coefficients $A, B; C^i, D^i, i = 1, 2, \ldots, d; Q, N$ and M may be random, with $A, B; C^i, D^i, i = 1, 2, \ldots, d; Q, N$ being $\mathscr{F}_t$-adapted and essentially bounded and M being $\mathscr{F}_T$-measurable and essentially bounded, Bismut (1976, 1978) used the stochastic maximum principle for optimal control and derived the following Riccati equation:

$$\begin{cases} -dK_t = [A_t^* K_t + K_t A_t + C_t^{i*} K_t C_t^i + C_t^{i*} L_t^i + L_t^i C_t^i \\ \qquad -\Psi_t(K_t, L_t)(N_t + D_t^{i*} K_t D_t^i)\Psi_t(K_t, L_t)] \, dt - L^i \, dW_t^i, \quad t \in [0, T); \\ K_T = M \end{cases} \quad (6)$$

where the function Ψ_t for $t \in [0, T]$ is defined as follows:

$$\Psi_t(K, L) := -(N_t + D_t^i K D_t^i)^{-1}(K B_t + C_t^{i*} K D_t^i + L^i D_t^i)^*, \forall\, K \in \mathscr{S}_+^n, \forall\, L$$
$$:= (L^1, \cdots, L^d) \in (R^{n \times n})^d.$$

Peng ([1992b]) used his stochastic Hamilton-Jacobi-Bellman equation to the SLQ problem and also derived the above equation. They both established the existence and uniqueness of an adapted solution of backward stochastic Riccati equation ([6]) when the function $\Psi_t(K, L)$ does not contain L. However, Bismut used the fixed-point method, and Peng ([1992b]) used Bellman's principle of quasilinearization and the method of monotone convergence. Neither methodology works for the general case of quadratic growth in the second unknown variable L in the drift of the stochastic equation. Bismut ([1976], [1978]) and Peng ([1999]) stated the general case as an open problem. By considering the stochastic equation for the inverse of K_t, Kohlmann and Tang ([2003a]) solved some particular cases where the function $\Psi_t(K, L)$ can depend on L. Tang ([2003]) finally solved the general case, using the method of stochastic flows.

In the general case, the optimal feedback coefficient $\Psi_t(K_t, L_t)$ at time t depends on L_t in a linear manner, which is in general not essentially bounded with respect to (t, ω). Kohlmann and Tang ([2003b]) observed that the stochastic integral process $\int_0^\cdot L_t^i\, dW_t^i$ is a BMO-martingale.

Indefinite SLQ Problem

Chen ([1985]) contains a theory of singular (the control weighting matrix vanishing in the quadratic cost functional) LQG control, which is a particular type of indefinite SLQ problems. In the deterministic linear-quadratic (LQ) control theory, the well posedness (i.e., the value function is finite on $[0, T] \times R^n$) of the problem suggests that the control weighting matrix N in the quadratic cost functional be positive definite. In the stochastic case, when N_t is slightly negative,

the SLQ may still be well posed if the control could also increase the intensity of the system noise. Peng ([1992a]) used an indefinite but well-posed SLQ problem to illustrate his new second-order stochastic maximum principle. Chen et al. ([1998]) gave a deeper study on this feature of the SLQ problem. Yong and Zhou ([1999]) gave a systematic account of the progress around in the indefinite SLQ problem.

Mean-Variance Hedging

In the theory of finance, Duffie and Richardson ([1991]) introduced the SLQ control model to hedge a contingent claim in an incomplete market. Schweizer ([1992]) developed a first framework for MV hedging, and then it was extended to a very general setting in Gouriéroux et al. ([1998]). Before 2000, the martingale method was used to solve the MV hedging problem. Kohlmann and Zhou ([2000]) began to use the standard SLQ theory to derive the optimal hedging strategy for a general contingent claim in a financial market of deterministic coefficients, and such a SLQ methodology was subsequently extended to very general settings for financial markets by Kohlmann and Tang ([2002], [2003b]), Bobrovnytska and Schweizer ([2004]), and Jeanblanc et al. ([2012]). See more detailed surveys on the literature by Pham ([2000]), Schweizer ([2010]), and Jeanblanc et al. ([2012]).

Summary and Future Directions

In comparison to the continuous-time deterministic LQ theory, the continuous-time SLQ theory has the following two striking features: An indefinite SLQ problem may be well posed, and the

S

optimal feedback coefficient may be unbounded due to its linear dependence on the martingale part L of the stochastic solution of the Riccati equation. Due to the second feature, the convergence of the sequence of successive approximations constructed via Bellman's quasilinearization still remains to be solved in the general case. This problem partially motivates Delbaen and Tang (2010) to study the regularity of unbounded stochastic differential equations and also may help to explain the necessity of rich studies on mean-variance hedging and closedness of stochastic integrals with respect to semimartingales (as in Delbaen et al. 1994, 1997) in various general settings.

Cross-References

▸ Stochastic Maximum Principle

Recommended Reading

The theory of SLQ control in various contexts is available in textbooks, monographs, or papers. Anderson and Moore (1971, 1989), Bensoussan (1992), and Chen (1985) include good accounts of the LQG control theory. Wonham (1970) includes a full introduction to the SLQ problem with deterministic piecewise continuous-time coefficients. Bismut (1978) gives a systematic and readable French introduction to SLQ problem with random coefficients. Yong and Zhou (1999) include an extensive discussion on the well-posed indefinite SLQ problem. Tang (2003) gives a complete solution of a general backward stochastic Riccati equation.

Bibliography

Anderson BDO, Moore JB (1971) Linear optimal control. Prentice-Hall, Englewood Cliffs

Anderson BDO, Moore JB (1989) Optimal control: linear quadratic methods. Prentice-Hall, Englewood Cliffs

Athans M (1971) The role and use of the stochastic linear-quadratic-Gaussian problem in control system design. IEEE Trans Autom Control AC-16(6):529–552

Bensoussan A (1992) Stochastic control of partially observable systems. Cambridge University Press, Cambridge

Bismut JM (1976) Linear quadratic optimal stochastic control with random coefficients. SIAM J Control Optim 14:419–444

Bismut JM (1978) Contrôle des systems linéaires quadratiques: applications de l'intégrale stochastique. In: Dellacherie C, Meyer PA, Weil M (eds) Séminaire de probabilités XII. Lecture Notes in Math 649. Springer, Berlin, pp 180–264

Bobrovnytska O, Schweizer M (2004) Mean-variance hedging and stochastic control: beyond the Brownian setting. IEEE Trans Autom Control 49:396–408

Chen H (1985) Recursive estimation and control for stochastic systems. Wiley, New York, pp 302–335

Chen S, Li X, Zhou X (1998) Stochastic linear quadratic regulators with indefinite control weight costs. SIAM J Control Optim 36:1685–1702

Delbaen F, Tang S (2010) Harmonic analysis of stochastic equations and backward stochastic differential equations. Probab Theory Relat Fields 146:291–336

Delbaen F et al (1994) Weighted norm inequalities and closedness of a space of stochastic integrals. C R Acad Sci Paris Sér I Math 319:1079–1081

Delbaen F et al (1997) Weighted norm inequalities and hedging in incomplete markets. Financ Stoch 1:181–227

Duffie D, Richardson HR (1991) Mean-variance hedging in continuous time. Ann Appl Probab 1:1–15

Florentin JJ (1961) Optimal control of continuous-time, Markov, stochastic systems. J Electron Control 10:473–488

Gouriéroux C, Laurent JP, Pham H (1998) Mean-variance hedging and numéraire. Math Financ 8:179–200

Jeanblanc M et al (2012) Mean-variance hedging via stochastic control and BSDEs for general semimartingales. Ann Appl Probab 22:2388–2428

Kohlmann M, Tang S (2002) Global adapted solution of one-dimensional backward stochastic Riccati equations, with application to the mean-variance hedging. Stoch Process Appl 97: 255–288

Kohlmann M, Tang S (2003a) Multidimensional backward stochastic Riccati equations and applications. SIAM J Control Optim 41:1696–1721

Kohlmann M, Tang S (2003b) Minimization of risk and linear quadratic optimal control theory. SIAM J Control Optim 42:1118–1142

Kohlmann M, Zhou XY (2000) Relationship between backward stochastic differential equations and stochastic controls: a linear-quadratic approach. SIAM J Control Optim 38:1392–1407

Peng S (1992a) New developments in stochastic maximum principle and related backward stochastic differential equations. In: Proceedings of the 31st conference on decision and control, Tucson, Dec 1992. IEEE, pp 2043–2047

Peng S (1992b) Stochastic Hamilton-Jacobi-Bellman equations. SIAM J Control Optim 30: 284–304

Peng S (1999) Open problems on backward stochastic differential equations. In: Chen S, Li X, Yong J, Zhou XY (eds) Control of distributed parameter and stochastic systems, IFIP, Hangzhou. Kluwer, pp 267–272

Pham H (2000) On quadratic hedging in continuous time. Math Methods Oper Res 51:315–339

Potter JE (1964) A guidance-navigation separation theorem. Experimental Astronomy Laboratory, Massachusetts Institute of Technology, Cambridge, Rep. RE-11, 1964

Schweizer M (1992) Mean-variance hedging for general claims. Ann Appl Probab 2:171–179

Schweizer M (2010) Mean-variance hedging. In: Cont R (ed) Encyclopedia of quantitative finance. Wiley, New York, pp 1177–1181

Tang S (2003) General linear quadratic optimal stochastic control problems with random coefficients: linear stochastic Hamilton systems and backward stochastic Riccati equations. SIAM J Control Optim 42:53–75

Tse E (1971) On the optimal control of stochastic linear systems. IEEE Trans Autom Control AC-16(6):776–785

Wonham WM (1967) Optimal stationary control of a linear system with state-dependent noise. SIAM J Control 5:486–500

Wonham WM (1968a) On the separation theorem of stochastic control. SIAM J Control 6:312–326

Wonham WM (1968b) On a matrix Riccati equation of stochastic control. SIAM J Control 6:681–697. Erratum (1969); SIAM J Control 7:365

Wonham WM (1970) Random differential equations in control theory. In: Bharucha-Reid AT (ed) Probabilistic methods in applied mathematics. Academic, New York, pp 131–212

Yong JM, Zhou XY (1999) Stochastic controls: Hamiltonian systems and HJB equations. Springer, New York

Stochastic Maximum Principle

Ying Hu
IRMAR, Université Rennes 1, Rennes Cedex, France

Abstract

The stochastic maximum principle (SMP) gives some necessary conditions for optimality for a stochastic optimal control problem. We give a summary of well-known results concerning stochastic maximum principle in finite-dimensional state space as well as some recent developments in infinite-dimensional state space.

Keywords

Introduction

The problem of finding sufficient conditions for optimality for a stochastic optimal control problem with finite-dimensional state equation had been well studied since the pioneering work of Bismut (1976, 1978). In particular, Bismut introduced linear backward stochastic differential equations (BSDEs) which have become an active domain of research since the seminal paper of Pardoux and Peng in 1990 concerning (nonlinear) BSDEs in Pardoux and Peng (1990).

The first results on SMP concerned only the stochastic systems where the control domain is convex or the diffusion coefficient does not contain control variable. In this case, only the first-order expansion is needed. This kind of SMP was developed by Bismut (1976, 1978), Kushner (1972), and Haussmann (1986). It is important to note that (Bismut 1978) introduced linear BSDE to represent the first-order adjoint process.

Peng made a breakthrough by establishing the SMP for the general stochastic optimal control problem where the control domain need not to be convex and the diffusion coefficient can contain the control variable. He solved this general case by introducing the second-order expansion and second-order BSDE. We refer to the book Yong and Zhou (1999) for the account of the theory of SMP in finite-dimensional spaces and describe Peng's SMP in the next section.

Despite the fact that the problem has been solved in complete generality more than 20 years ago, the infinite-dimensional case still has important open issues both on the side of the generality of the abstract model and on the side of its applicability to systems modeled by stochastic partial differential equations (SPDEs). The last section is devoted to the recent development of SMP in infinite-dimensional space.

S

Statement of SMP

Formulation of Problem

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, on which an m-dimensional Brownian motion W is given. Let $\{\mathcal{F}_t\}_{t \geq 0}$ be the natural completed filtration of W.

We consider the following stochastic controlled system:

$$dx(t) = b(x(t), u(t))dt + \sigma(x(t), u(t))dW(t),$$
$$x(0) = x_0, \qquad (1)$$

with the cost functional

$$J(u(\cdot)) = \mathbb{E}\left\{\int_0^T f(x(t), u(t))dt + h(x(T))\right\}. \qquad (2)$$

In the above, b, σ, f, h are given functions with appropriate dimensions. (U, d) is a separable metric space.

We define

$$\mathcal{U} = \{u : [0, T] \times \Omega$$
$$\to U \mid u \text{ is } \{\mathcal{F}_t\}_{t \geq 0} - \text{ adapted }\}. \quad (3)$$

The optimal problem is: Minimize $J(u(\cdot))$ over $\mathcal{U}$.

Any $\bar{u} \in \mathcal{U}$ satisfying

$$J(\bar{u}) = \inf_{u \in \mathcal{U}} J(u) \qquad (4)$$

is called an optimal control. The corresponding $\bar{x}$ and $(\bar{x}, \bar{u})$ is called an optimal state process/trajectory and optimal pair, respectively.

In this section, we assume the following standard hypothesis:

Hypothesis 1 *1. The functions $b : \mathbb{R}^n \times U \mapsto \mathbb{R}^n$, $\sigma = (\sigma^1, \cdots, \sigma^m) : \mathbb{R}^n \times U \mapsto \mathbb{R}^{n \times m}$, $f : \mathbb{R}^n \times U \mapsto \mathbb{R}$ and $h : \mathbb{R}^n \mapsto \mathbb{R}$ are measurable functions.*
2. For $\varphi = b, \sigma^j, j = 1, \cdots, m, f$, the functions $x \mapsto \varphi(x, u)$ and $x \mapsto h(x)$ are C^2, denoted φ_x and φ_{xx} (respectively, h_x and h_{xx}), which are also continuous functions of (x, u).
3. There exists a constant $K > 0$ such that

$$|\varphi_x| + |\varphi_{xx}| + |h_x| + |h_{xx}| \leq K,$$

and

$$|\varphi| + |h| \leq K(1 + |x| + |u|).$$

Adjoint Equations

Let us first introduce the following backward stochastic differential equations (BSDEs).

$$dp(t) = -\{b_x(\bar{x}(t), \bar{u}(t))^T p(t) \qquad (5)$$
$$+ \sum_{j=1}^m \sigma_x^j(\bar{x}(t), \bar{u}(t))^T q_j(t)$$
$$- f_x(\bar{x}(t), \bar{u}(t))\}dt + q(t)dW(t),$$
$$p(T) = -h_x(\bar{x}(T)).$$

The solution (p, q) to the above BSDE (first-order BSDE) is called the first-order adjoint process.

$$dP(t) = -\{b_x(\bar{x}(t), \bar{u}(t))^T P(t) + P(t)b_x(\bar{x}(t), \bar{u}(t)) + \sum_{j=1}^m \sigma_x^j(\bar{x}(t), \bar{u}(t))^T P(t)\sigma_x^j(\bar{x}(t), \bar{u}(t))$$

$$+ \sum_{j=1}^m \{\sigma_x^j(\bar{x}(t), \bar{u}(t))^T Q_j(t) + Q_j(t)\sigma_x^j(\bar{x}(t), \bar{u}(t))\}$$

$$+ H_{xx}(\bar{x}(t), \bar{u}(t), p(t), q(t))\}dt + \sum_{j=1}^m Q_j(t)dW^j(t), \qquad (6)$$

$$P(T) = -h_{xx}(\bar{x}(T)),$$

where the Hamiltonian H is defined by

$$H(x, u, p, q) = \langle p, b(x, u) \rangle$$
$$+ \text{tr}[q^T \sigma(x, u)] - f(x, u). \quad (7)$$

The solution (P, Q) to the above BSDE (second-order BSDE) is called the second-order adjoint process.

Stochastic Maximum Principle

Let us now state the stochastic maximum principle.

Theorem 1 *Let* $(\bar{x}, \bar{u})$ *be an optimal pair of problem. Then there exist a unique couple* (p, q) *satisfying* (5) *and a unique couple* (P, Q) *satisfying* (6)*, and the following maximum condition holds:*

$$H(\bar{x}(t), \bar{u}(t), p(t), q(t)) - H(\bar{x}(t), u, p(t), q(t))$$
$$- \frac{1}{2} tr(\{\sigma(\bar{x}(t), \bar{u}(t)) - \sigma(\bar{x}(t), u)\}^T P(t) \{\sigma(\bar{x}(t), \bar{u}(t)) - \sigma(\bar{x}(t), u)\}) \geq 0. \quad (8)$$

SMP in Infinite-Dimensional Space

The problem of finding sufficient conditions for optimality for a stochastic optimal control problem with infinite-dimensional state equation, along the lines of the Pontryagin maximum principle, was already addressed in the early 1980s in the pioneering paper (Bensoussan 1983).

Whereas the Pontryagin maximum principle for infinite-dimensional stochastic control problems is a well-known result as far as the control domain is convex (or the diffusion does not depend on the control; see Bensoussan 1983; Hu and Peng 1990), for the general case (that is when the control domain need not be convex and the diffusion coefficient can contain a control variable), existing results are limited to abstract evolution equations under assumptions that are not satisfied by the large majority of concrete SPDEs.

The technical obstruction is related to the fact that (as it was pointed out in Peng 1990) if the control domain is not convex, the optimal control has to be perturbed by the so-called spike variation. Then if the control enters the diffusion, the irregularity in time of the Brownian trajectories imposes to take into account a second variation process. Thus, the stochastic maximum principle has to involve an adjoint process for the second variation. In the finite-dimensional case, such a process can be characterized as the solution

of a matrix-valued backward stochastic differential equation (BSDE), while in the infinite-dimensional case, the process naturally lives in a non-Hilbertian space of operators and its characterization is much more difficult. Moreover, the applicability of the abstract results to concrete controlled SPDEs is another delicate step due to the specific difficulties that they involve such as the lack of regularity of Nemytskii-type coefficients in L^p spaces.

Concerning results on the infinite-dimensional stochastic Pontryagin maximum principle, as we already mentioned, in Bensoussan (1983) and Hu and Peng (1990), the case of diffusion independent on the control is treated (with the difference that in Hu and Peng (1990) a complete characterization of the adjoint to the first variation as the unique mild solution to a suitable BSDE is achieved).

The paper Tang and Li (1994) is the first one in which the general case is addressed with, in addition, a general class of noises possibly with jumps. The adjoint process of the second variation $(P_t)_{t \in [0,T]}$ is characterized as the solution of a BSDE in the (Hilbertian) space of Hilbert-Schmidt operators. This forces to assume a very strong regularity on the abstract state equation and control functional that prevents application of the results in Tang and Li (1994) to SPDEs.

Then in the papers by Fuhrman et al. (2012, 2013), the state equation is formulated, only in a semiabstract way in order, on one side, to cope

with all the difficulties carried by the concrete nonlinearities and, on the other, to take advantage of the regularizing properties of the leading elliptic operator.

Recently in Lü and Zhang (2012), P_t was characterized as "transposition solution" of a backward stochastic evolution equation in $\mathcal{L}(L^2(\mathcal{O}))$. Coefficients are required to be twice Fréchet differentiable as operators in $L^2(\mathcal{O})$. Finally, even more recently in a couple of preprints (Du and Meng (2012, 2013)), the process P_t is characterized in a similar way as it is in Fuhrman et al. (2012, 2013). Roughly speaking it is characterized as a suitable stochastic bilinear form. As it is the case in Lü and Zhang (2012), in Du and Meng (2012, 2013) as well, the regularity assumptions on the coefficients are too restrictive to apply directly the results in Lü and Zhang (2012), Du and Meng (2012, 2013) to controlled SPDEs.

Cross-References

▶ Backward Stochastic Differential Equations and Related Control Problems
▶ Numerical Methods for Continuous-Time Stochastic Control Problems
▶ Stochastic Adaptive Control
▶ Stochastic Linear-Quadratic Control

Bibliography

Bensoussan A (1983) Stochastic maximum principle for distributed parameter systems. J Frankl Inst 315(5–6):387–406
Bismut JM (1976) Linear quadratic optimal stochastic control with random coefficients. SIAM J Control Optim 14(3):419–444
Bismut JM (1978) An introductory approach to duality in optimal stochastic control. SIAM Rev 20(1):62–78
Du K, Meng Q (2012) Stochastic maximum principle for infinite dimensional control systems. arXiv:1208.0529
Du K, Meng Q (2013) A maximum principle for optimal control of stochastic evolution equations. SIAM J Control Option 51(4):4343–4362
Fuhrman M, Hu Y, Tessitore G (2012) Stochastic maximum principle for optimal control of SPDEs. C R Math Acad Sci Paris 350(13–14):683–688
Fuhrman M, Hu Y, Tessitore G (2013) Stochastic maximum principle for optimal control of SPDEs. Appl Math Optim 68(2):181–217
Haussmann UG (1986) A stochastic maximum principle for optimal control of diffusions. Pitman research notes in mathematics series, vol 151. Longman Scientific & Technical, Harlow/Wiley, New York
Hu Y, Peng S (1990) Maximum principle for semilinear stochastic evolution control systems. Stoch Stoch Rep 33(3–4):159–180
Kushner HJ (1972) Necessary conditions for continuous parameter stochastic optimization problems. SIAM J Control 10:550–565
Lü Q, Zhang X (2012) General Pontryagin-type stochastic maximum principle and backward stochastic evolution equations in infinite dimensions. arXiv:1204.3275
Pardoux E, Peng S (1990) Adapted solution of a backward stochastic differential equation. Syst Control Lett 14(1):55–61
Peng S (1990) A general stochastic maximum principle for optimal control problems. SIAM J Control Optim 28(4):966–979
Tang S, Li X (1994) Maximum principle for optimal control of distributed parameter stochastic systems with random jumps. In: Markus L, Elworthy KD, Everitt WN, Lee EB (eds) Differential equations, dynamical systems, and control science. Lecture notes in pure and applied mathematics, vol 152. Dekker, New York, pp 867–890
Yong J, Zhou XY (1999) Stochastic controls: Hamiltonian systems and HJB equations. Applications of mathematics, vol 43. Springer, New York

Stochastic Model Predictive Control

Basil Kouvaritakis and Mark Cannon
Department of Engineering Science, University of Oxford, Oxford, UK

Abstract

Model predictive control (MPC) is a control strategy that has been used successfully in numerous and diverse application areas. The aim of the present entry is to discuss how the basic ideas of MPC can be extended to problems involving random model uncertainty with known probability distribution. We discuss cost indices, constraints, closed-loop properties, and implementation issues.

Keywords

Mean-square stability; Recursive feasibility; Stochastic Lyapunov function

Introduction

Stochastic model predictive control (SMPC) refers to a family of numerical optimization strategies for controlling stochastic systems subject to constraints on the states and inputs of the controlled system. In this approach, future performance is quantified using a cost function evaluated along predicted state and input trajectories. This leads to a stochastic optimal control problem, which is solved numerically to determine an optimal open-loop control sequence or alternatively a sequence of feedback control laws. In MPC, only the first element of this optimal sequence is applied to the controlled system, and the optimal control problem is solved again at the next sampling instant on the basis of updated information on the system state. The numerical nature of the approach makes it applicable to systems with nonlinear dynamics and constraints on states and inputs, while the repeated computation of optimal predicted trajectories introduces feedback to compensate for the effects of uncertainty in the model.

Robust MPC (RMPC) tackles problems with hard state and input constraints, which are to be satisfied for all realizations of model uncertainty. However, RMPC is too conservative in many applications and stochastic MPC (SMPC) provides less conservative solutions by handling a wider class of constraints which are to be satisfied in mean or with a specified probability. This is achieved by taking explicit account of the probability distribution of the stochastic model uncertainty in the optimization of predicted performance. Constraints limit performance and an advantage of MPC is that it allows systems to operate close to constraint boundaries. Stochastic MPC is similarly advantageous when model uncertainty is stochastic with known probability distribution and the constraints are probabilistic in nature.

Applications of SMPC have been reported in diverse fields, including finance and portfolio management, risk management, sustainable development policy assessment, chemical and process industries, electricity generation and distribution, building climate control, andtelecommunications network traffic control. This entry aims to summarize the theoretical framework underlying SMPC algorithms.

Stochastic MPC

Consider a system with discrete time model

$$x^+ = f(x, u, w) \qquad (1)$$

$$z = g(x, u, v) \qquad (2)$$

where $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ are the system state and control input and x^+ is the successor state (i.e., if x_i is the state at time i, then $x^+ = x_{i+1}$ is the state at time $i + 1$). Inputs $w \in \mathbb{R}^{n_w}$ and $v \in \mathbb{R}^{n_v}$ are exogenous disturbances with unknown current and future values but known probability distributions, and $z \in \mathbb{R}^{n_z}$ is a vector of output variables that are subject to constraints.

The optimal control problem that is solved online at each time step in SMPC is defined in terms of a performance index $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ evaluated over a future horizon of N time steps. Typically in SMPC $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ is a quadratic function of the following form (in which $\|x\|_Q^2 = x^T Q x$)

$$J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}) = \sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2) + V_f(\hat{x}_N) \qquad (3)$$

for positive definite matrices Q and R, and a terminal cost $V_f(x)$ defined as discussed in section "Stability and Convergence." Here $\hat{\mathbf{u}} := \{\hat{u}_0, \ldots, \hat{u}_{N-1}\}$ is a postulated sequence of control inputs and $\hat{\mathbf{x}}(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}) := \{\hat{x}_0, \ldots, \hat{x}_N\}$ is the corresponding sequence of states such that $\hat{x}_i$ is the solution of (1) at time i with initial state $\hat{x}_0 = x$, for a given sequence of disturbance inputs $\hat{\mathbf{w}} := \{\hat{w}_0, \ldots, \hat{w}_{N-1}\}$. Since $\hat{\mathbf{w}}$ is a random sequence, $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ is a random variable, and the optimal control problem is therefore formulated as the minimization of a cost $V_N(x, \hat{\mathbf{u}})$ derived from $J_N(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$ under specific assumptions on $\hat{\mathbf{w}}$. Common definitions of $V_N(x, \hat{\mathbf{u}})$ are as follows.

S

(a) Expected value cost:

$$V_N(x, \hat{\mathbf{u}}) := \mathbb{E}_x(J(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}))$$

where $\mathbb{E}_x(\cdot)$ denotes the conditional expectation of a random variable $(\cdot)$ given the model state x.

(b) Worst-case cost, assuming $\hat{w}_i \in \mathcal{W}$ for all i with probability 1, for some compact set $\mathcal{W} \subset \mathbb{R}^{n_w}$:

$$V_N(x, \hat{\mathbf{u}}) := \max_{\hat{\mathbf{w}} \in \mathcal{W}^N} J(x, \hat{\mathbf{u}}, \hat{\mathbf{w}}).$$

(c) Nominal cost, assuming $\hat{w}_i$ is equal to some nominal value, e.g., if $\hat{w}_i = 0$ for all i, then

$$V_N(x, \hat{\mathbf{u}}) := J(x, \hat{\mathbf{u}}, \mathbf{0}),$$

where $\mathbf{0} = \{0, \ldots, 0\}$.

The minimization of $V_N(x, \hat{\mathbf{u}})$ is performed subject to constraints on the sequence of outputs $\hat{z}_i := g(\hat{x}_i, \hat{u}_i, \hat{v}_i)$, $i \geq 0$. These constraints may be formulated in various ways, summarized as follows, where for simplicity we assume $n_z = 1$.

(A) Expected value constraints: for all i,

$$\mathbb{E}_x(\hat{z}_i) \leq 1.$$

(B) Probabilistic constraints pointwise in time:

$$\Pr_x(\hat{z}_i \leq 1) \geq p,$$

for all i and for a given probability p.

(C) Probabilistic constraints over a future horizon:

$$\Pr_x(\hat{z}_i \leq 1, \ i = 0, 1, \ldots, N) \geq p$$

for a given probability p.

In (B) and (C), $\Pr_x(\mathcal{A})$ represents the conditional probability of an event $\mathcal{A}$ that depends on the sequence $\hat{\mathbf{x}}(x, \hat{\mathbf{u}}, \hat{\mathbf{w}})$, given that the initial model state is $\hat{x}_0 = x$; for example the probability $\Pr_x(\hat{z}_i \leq 1)$ depends on the distribution of $\{\hat{w}_0, \ldots, \hat{w}_{i-1}, \hat{v}_i\}$.

The important special case of state constraints can also be handled by (A)–(C) through appropriate choice of the function $g(x, u, v)$. For example the constraint $\Pr_x(h(x) \leq 1) \geq p$, for a given function $h : \mathbb{R}^n \to \mathbb{R}$, can be expressed in the form (B) with $z = g(x, u, v) := h(f(x, u, w))$ and $v := w$ in (2).

In common with other receding horizon control strategies, SMPC is implemented via the following algorithm. At each discrete time step:

 (i) Minimize the cost index $V_N(x, \hat{\mathbf{u}})$ over $\hat{\mathbf{u}}$ subject to the constraints on $\hat{z}_i$, $i \geq 0$, given the current system state x.

(ii) Apply the control input $u = \hat{u}_0^*(x)$ to the system, where $\hat{\mathbf{u}}^*(x) = \{\hat{u}_0^*(x), \ldots, \hat{u}_{N-1}^*(x)\}$ is the minimizing sequence given x.

If the system dynamics (1) are unstable, then performing the optimization in step (i) directly over future control sequences can result in a small set of feasible states x. To avoid this difficulty the elements of the control sequence $\hat{\mathbf{u}}$ are usually expressed in the form $\hat{u}_i = u_T(\hat{x}_i) + s_i$, where $u_T(x)$ is a locally stabilizing feedback law, and $\{s_0, \ldots, s_{N-1}\}$ are optimization variables in step (i).

Constraints and Recursive Feasibility

The constraints in (B) and (C) include hard constraints ($p = 1$) as a special case, but in general the conditions (A)–(C) represent soft constraints that are not required to hold for all realizations of model uncertainty. However, these constraints can only be satisfied if the state belongs to a subset of state space, and the requirement (common in MPC) that the optimization in step (i) of the SMPC algorithm should remain feasible if it is initially feasible therefore implies additional constraints. For example, the condition $\Pr_x(\hat{z}_0 \leq 1) \geq p$ can be satisfied only if x belongs to the set for which there exists $\hat{u}_0$ such that $\Pr_x(g(x, \hat{u}_0, \hat{v}_0) \leq 1) \geq p$. Hence, soft constraints implicitly impose hard constraints on the model state.

SMPC algorithms typically handle the conditions relating to feasibility of constraint sets in

one of two ways. Either the SMPC optimization is allowed to become infeasible (often with penalties on constraint violations included in the cost index), or conditions ensuring robust feasibility of the SMPC optimization at all future times are imposed as extra constraints in the SMPC optimization.

The first of these approaches has been used in the context of constraints (C) imposed over a horizon, for which conditions ensuring future feasibility are generally harder to characterize in terms of algebraic conditions on the model state than (A) or (B). A disadvantage of this approach is that the closed-loop system may not satisfy the required soft constraints, even if these constraints are feasible when applied to system trajectories predicted at initial time.

The second approach treats conditions for feasibility as hard constraints and hence requires a guarantee of recursive feasibility, namely, that the SMPC optimization must remain feasible for the closed-loop system if it is feasible initially. This can be achieved by requiring, similarly to RMPC, that the conditions for feasibility of the SMPC optimization problem should be satisfied for all realizations of the sequence $\hat{\mathbf{w}}$. For example, for given $\hat{x}_0 = x$, there exists $\hat{\mathbf{u}}$ satisfying that the conditions of (B) if

$$\Pr_{\hat{x}_i}(g(\hat{x}_i, \hat{u}_i, \hat{v}_i) \leq 1) \geq p, \, i = 0, 1, \dots \quad (4a)$$

$$\hat{x}_i \in X \, \forall \{\hat{w}_0, \dots, \hat{w}_{i-1}\} \in \mathcal{W}^i, \, i = 1, 2, \dots \quad (4b)$$

where X is the set

$$X = \{x : \exists u \text{ such that } \Pr_x(g(x, u, v) \leq 1) \geq p\}.$$

Furthermore, an SMPC optimization that includes the constraints of (4) must remain feasible at subsequent times (since (4) ensures the existence of $\hat{\mathbf{u}}^+$ such that each element of $\hat{\mathbf{x}}(f(x, \hat{u}_0, \hat{w}_0), \hat{\mathbf{u}}^+, \hat{\mathbf{w}}^+)$ lies in X for all $\hat{w}_0 \in \mathcal{W}$ and all $\hat{\mathbf{w}}^+ \in \mathcal{W}^N$).

Satisfaction of (4) at each time step i on the infinite horizon $i \geq N$ can be ensured through a finite number of constraints by introducing constraints on the N-step-ahead state $\hat{x}_N$. This approach uses a fixed feedback law, $u_T(x)$, to define a postulated input sequence after the initial N-step horizon via $\hat{u}_i = u_T(\hat{x}_i)$ for all $i \geq N$. The constraints of (4) are necessarily satisfied for all $i \geq N$ if a constraint

$$\hat{x}_N \in X_T$$

is imposed, where X_T is robustly positively invariant with probability 1 under $u_T(x)$, i.e.

$$f(x, u_T(x), w) \in X_T, \, \forall x \in X_T, \, \forall w \in \mathcal{W}, \quad (5)$$

and furthermore the constraint $\Pr_x(z \leq 1) \geq p$ is satisfied at each point in X_T under $u_T(x)$, i.e.,

$$\Pr_x(g(x, u_T(x), v) \leq 1) \geq p, \, \forall x \in X_T.$$

Although the recursively feasible constraints (4) account robustly for the future realizations of the unknown parameter w in (1), the key difference between SMPC and RMPC is that the conditions in (4) depend on the probability distribution of the parameter v in (2). It also follows from the necessity of hard constraints for feasibility that the distribution of w must in general have finite support in order that feasibility can be guaranteed recursively. On the other hand the support of v in the definition of z may be unbounded (an important exception being the case of state constraints in which $v = w$).

Stability and Convergence

This section outlines the stability properties of SMPC strategies based on cost indices (a)–(c) of section "Stochastic MPC" and related variants. We use $V_N^*(x) = V_N(x, \hat{\mathbf{u}}^*(x))$ to denote the optimal value of the SMPC cost index, and X_T denotes a subset of state space satisfying the robust invariance condition (5). We also denote the solution at time i of the system (1) with initial state $x_0 = x$ and under a given feedback control law $u = \kappa(x)$ and disturbance sequence $\mathbf{w} = \{w_0, w_1, \dots\}$ as $x_i(x, \kappa, \mathbf{w})$.

The expected value cost index in (a) results in mean-square stability of the closed-loop system provided the terminal term $V_f(x)$ in (3) satisfies

$$\mathbb{E}_x V_f(f(x, u_T(x), w) \leq V_f(x) - \|x\|_Q^2$$
$$- \|u_T(x)\|_R^2$$

for all x in the terminal set X_T. The optimal cost is then a stochastic Lyapunov function satisfying

$$\mathbb{E}_x V_N^*(f(x, \hat{u}_0^*(x), w)) \leq V_N^*(x) - \|x\|_Q^2$$
$$- \|\hat{u}_0^*(x)\|_R^2.$$

For positive definite Q this implies the closed-loop system under the SMPC law is mean-square stable, so that $x_i(x, \hat{u}_0^*, \mathbf{w}) \to 0$ as $i \to \infty$ with probability 1 for any feasible initial condition x. For the case of systems (1) subject to additive disturbances, the modified cost

$$V_N(x, \hat{\mathbf{u}}) := \mathbb{E}_x \left[\sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2 - l_{ss}) + V_f(\hat{x}_N) \right]$$

where $l_{ss} := \lim_{i \to \infty} \mathbb{E}_x(\|x_i(x, u_T, \mathbf{w})\|_Q^2 + \|u_i\|_R^2)$ under $u_i = u_T(x_i)$ results in the asymptotic bound

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \mathbb{E}_x(\|x_i(x, \hat{u}_0^*, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq l_{ss}$$

along the closed-loop trajectories of (1) under the SMPC law $u_i = \hat{u}_0^*(x_i)$, for any feasible initial condition x.

For the worst-case cost (b), if $V_f(x)$ is designed as a control Lyapunov function for (1), with

$$V_f(f(x, u_T(x), w) \leq V_f(x) - \|x\|_Q^2 - \|u_T(x)\|_R^2$$

for all $w \in \mathcal{W}$ and all $x \in X_T$, then $V_N^*(x)$ is a Lyapunov function satisfying

$$V_N^*(f(x, \hat{u}_0^*(x), w) \leq V_N^*(x) - \|x\|_Q^2 - \|\hat{u}_0^*(x)\|_R^2$$

for all $w \in \mathcal{W}$, implying $x = 0$ is an asymptotically stable equilibrium of (1) under the SMPC law $u = \hat{u}_0^*(x)$. Clearly the system model (1) cannot be subject to unknown additive disturbances in this case. However, for the case in which the system (1) is subject to additive disturbances, a variant of this approach uses a modified cost which is equal to zero inside some set of states, leading to asymptotic stability of this set rather than an equilibrium point. Also in the context of additive disturbances, an alternative approach uses an $\mathcal{H}_\infty$-type cost,

$$V_N(x, \hat{\mathbf{u}}) := \max_{\hat{\mathbf{w}} \in \mathcal{W}^N} \left[\sum_{i=0}^{N-1} (\|\hat{x}_i\|_Q^2 + \|\hat{u}_i\|_R^2 - \gamma^2 \|\hat{w}_i\|^2) + V_f(\hat{x}_N) \right]$$

for which the closed-loop trajectories of (1) under the associated SMPC law $u_i = \hat{u}_0^*(x_i)$ satisfy

$$\sum_{i=0}^{\infty} (\|x_i(x, \hat{u}_0^*, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq \gamma^2$$
$$\sum_{i=0}^{\infty} \|w_i\|^2 + V_N^*(x_0)$$

provided $V_f(f(x, u_T(x), w)) \leq V_f(x) - (\|x\|_Q^2 + \|u_T(x)\|_R^2 - \gamma^2 \|w\|^2)$ for all $w \in \mathcal{W}$ and $x \in X_T$.

Algorithms employing the nominal cost (c) typically rely on the existence of a feedback law $u_T(x)$ such that the system (1) satisfies, in the absence of constraints and under $u_i = u_T(x_i)$, an input-to-state stability (ISS) condition of the form

$$\sum_{i=0}^{\infty} (\|x_i(x, u_T, \mathbf{w})\|_Q^2 + \|u_i\|_R^2) \leq \gamma^2 \sum_{i=0}^{\infty} \|w_i\|^2 + \beta \tag{6}$$

for some γ and $\beta > 0$. If $V_f(x)$ satisfies

$$V_f(f(x, u_T(x), 0)) \leq V_f(x) - (\|x\|_Q^2 + \|u_T(x)\|_R^2)$$

for all $x \in X_T$, then the closed-loop system under SMPC with the nominal cost (c) satisfies an ISS condition with the same gain γ as the unconstrained case (6) but a different constant β.

Implementation Issues

In general stochastic MPC algorithms require more computation than their robust counterparts because of the need to determine the probability distributions of future states. An important exception is the case of linear dynamics and purely additive disturbances, for which the model (1)–(2) becomes

$$x^+ = Ax + Bu + w \qquad (7)$$

$$z = Cx + Du + v \qquad (8)$$

where A, B, C, D are known matrices. In this case the expected value constraints (A) and probabilistic constraints (B), as well as hard constraints that ensure future feasibility of the SMPC optimization in each case, can be invoked nonconservatively through tightened constraints on the expectations of future states. Furthermore, the required degree of tightening can be computed off-line using numerical integration of probability distributions or using random sampling techniques, and the online computational load is similar to MPC with no model uncertainty.

The case in which the matrices A, B, C, D in the model (7)–(8) depend on unknown stochastic parameters is more difficult because the predicted states then involve products of random variables. An effective approach to this problem uses a sequence of sets (known as a tube) to recursively bound the sequence of predicted states via one step-ahead set inclusion conditions. By using polytopic bounding sets that are defined as the intersection of a fixed number of half-spaces, the complexity of these tubes can be controlled by the designer, albeit at the expense of conservative inclusion conditions. Furthermore, an application of Farkas' Lemma allows these sets

to be computed online through linear conditions on optimization variables.

Random sampling techniques developed for general stochastic programming problems provide effective means of handling the soft constraints arising in SMPC. These techniques use finite sets of discrete samples to represent the probability distributions of model states and parameters. Furthermore bounds are available on the number of samples that are needed in order to meet specified confidence levels on the satisfaction of constraints. Probabilistic and expected value constraints can be imposed using random sampling, and this approach has also been applied to the case of probabilistic constraints over a horizon (C) through a scenario-based optimization approach.

Summary and Future Directions

This entry describes how the ideas of MPC and RMPC can be extended to the case of stochastic model uncertainty. Crucial in this development is the assumption that the uncertainty has bounded support, which allows the assertion of recursive feasibility of the SMPC optimization problem. For simplicity of presentation we have considered the case of full-state feedback. However, stochastic MPC can also be applied to the output feedback case using a state estimator if the probability distributions of measurement and estimation noise are known.

An area of future development is optimization over sequences of feedback policies. Although an observer at initial time cannot know the future realizations of random uncertainty, information on $\hat{x}_i$ will be available to the controller i-steps ahead, and, as mentioned in section "Stochastic MPC" in the context of feasible initial condition sets, $\hat{u}_i$ must therefore depend on $\hat{x}_i$. In general the optimal control decision is of the form $\hat{u}_i = \mu_i(\hat{x}_i)$ where $\mu_i(\cdot)$ is a feedback policy. This implies optimization over arbitrary feedback policies, which is generally considered to be intractable since the required online computation grows exponentially with the

horizon N. However, approximate approaches to this problem have been suggested which optimize over restricted classes of feedback laws, and further developments in this respect are expected in the future.

Cross-References

▶ Distributed Model Predictive Control
▶ Economic Model Predictive Control
▶ Nominal Model-Predictive Control
▶ Robust Model-Predictive Control
▶ Tracking Model Predictive Control

Recommended Reading

A historical perspective on SMPC is provided by Åström and Wittenmark (1973), Charnes and Cooper (1963), and Schwarm and Nikolaou (1999). A treatment of constraints stated in terms of expected values can be found, for example, in Primbs and Sung (2009). Probabilistic constraints and the conditions for recursive feasibility can be found in Kouvaritakis et al. (2010) for the additive case, whereas the general case of multiplicative and additive uncertainty is described in Evans et al. (2012), which uses random sampling techniques. Random sampling techniques were developed for random convex programming (Calafiore and Campi 2005) and were used in a scenario-based approach to predictive control in Calafiore and Fagiano (2013). An output feedback SMPC strategy incorporating state estimation is described in Cannon et al. (2012).

The use of the expectation of a quadratic cost and associated mean-square stability results are discussed in Lee and Cooley (1998). Robust stability results for MPC based on worst-case costs are given by Lee and Yu (1997) and Mayne et al. (2005). Input-to-state stability of MPC based on a nominal cost is discussed in Marruedo et al. (2002).

Descriptions of SMPC based on closed-loop optimization can be found in Lee and Yu (1997) and Stoorvogel et al. (2007). These algorithms are computationally intensive and approximate solutions can be found by restricting the class of closed-loop predictions as discussed, for example, in van Hessem and Bosgra (2002) and Primbs and Sung (2009).

Bibliography

Åström KJ, Wittenmark B (1973) On self tuning regulators. Automatica 9(2):185–199

Calafiore GC, Campi MC (2005) Uncertain convex programs: randomized solutions and confidence levels. Math Program 102(1):25–46

Calafiore GC, Fagiano L (2013) Robust model predictive control via scenario optimization. IEEE Trans Autom Control 58(1):219–224

Cannon M, Cheng Q, Kouvaritakis B, Rakovic SV (2012) Stochastic tube MPC with state estimation. Automatica 48(3):536–541

Charnes A, Cooper WW (1963) Deterministic equivalents for optimizing and satisficing under chance constraints. Oper Res 11(1):19–39

Evans M, Cannon M, Kouvaritakis B (2012) Robust MPC for linear systems with bounded multiplicative uncertainty. In: IEEE conference on decision and control, Maui, pp 248–253

Kouvaritakis B, Cannon M, Raković SV, Cheng Q (2010) Explicit use of probabilistic distributions in linear predictive control. Automatica 46(10):1719–1724

Lee JH, Cooley BL (1998) Optimal feedback control strategies for state-space systems with stochastic parameters. IEEE Trans Autom Control 43(10):1469–1475

Lee JH, Yu Z (1997) Worst-case formulations of model predictive control for systems with bounded parameters. Automatica 33(5):763–781

Marruedo DL, Alamo T, Camacho EF (2002) Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: IEEE conference on decision and control, Las Vegas, pp 4619–4624

Mayne DQ, Seron MM, Raković SV (2005) Robust model predictive control of constrained linear systems with bounded disturbances. Automatica 41(2):219–224

Primbs JA, Sung CH (2009) Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. IEEE Trans Autom Control 54(2):221–230

Schwarm AT, Nikolaou M (1999) Chance-constrained model predictive control. AIChE J 45(8):1743–1752

Stoorvogel AA, Weiland S, Batina I (2007) Model predictive control by randomized algorithms for systems with constrained inputs and stochastic disturbances. http://wwwhome.math.utwente.nl/~stoorvogelaa/subm01.pdf

van Hessem DH, Bosgra OH (2002) A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: IEEE conference on decision and control, Las Vegas, pp 4643–4648

Stock Trading via Feedback Control

B. Ross Barmish[1] and James A. Primbs[2]
[1]University of Wisconsin, Madison, WI, USA
[2]University of Texas at Dallas, Richardson, TX, USA

Abstract

This article covers stock trading from a feedback control point of view. To this end, the mechanics and practical considerations associated with the use of feedback-based algorithms are explained for both real-world trading and scenarios involving numerical simulation.

Keywords and Phrases

Feedback Control; Finance; Model-Free; Stock Trading

Introduction

Stock trading involves the purchase and sale of *shares* of ownership in public companies by an individual or entity such as a pension fund, mutual fund, hedge fund, or endowment. These shares are typically traded in markets, such as the New York Stock Exchange and the NASDAQ, with the trader's goal generally being to increase wealth. The words *feedback control* in the title of this article broadly refer to the use of information such as prices, profits and losses which becomes available to the trader over time and is used to make purchase and sales decisions according to some set of rules. That is, the size of the stock position being held varies with time. The mapping from information to the investment level is called the *feedback law* and is typically described with a closed-loop configuration and

classical algorithms which come from the body of research called control theory; e.g., see Astrom and Murray (2008).

For simplicity, in this article, we restrict attention to trading a single stock while noting that the concepts described herein are readily modified to address the multi-stock case, i.e., a *portfolio*. To our knowledge, the basic idea of viewing portfolios in a control-theoretic setting goes back to Merton (1969) where optimal control concepts are explicitly used; see also Samuelson (1969) where a less general formulation is considered. Whereas the theoretical foundations in their work rely on idealized assumptions such as "frictionless markets" and "continuous trading," the main objective in this article is to describe the practical considerations and complexities which arise in real-world stock trading via feedback control and associated simulations. That is, the exposition to follow includes no significant idealizing assumptions and emphasizes implementation issues and constraints which are encountered by the practitioner; i.e., the purpose of this article is to describe trading mechanics in a feedback context. Hence, when we define a trading strategy in the sequel, we include no significant discussion of performance metrics related to risk and return; the reader is referred to the book by Luenberger (1998) for coverage of these topics.

Feedback Versus Open-Loop Control

We first elaborate on the definitions above by pointing out the distinction between trading a stock via feedback control and its alternative, "open-loop control." This is done via simple examples: Suppose an investor buys $1,000 of stock at time $t = 0$ with the a priori plan to make no changes in this position until some prespecified future time $t = T$. Then, this *buy-and-hold* trading strategy falls within the realm of open-loop control. If instead this same investor adds $1,000 to the position every month, then this type of *dollar-cost averaging* strategy would still fall into the open-loop category. That is, in both scenarios, no information is being used to modify the stock position over time. Finally, suppose this same investor makes a $1,000 purchase only at the end

of those months over which the account value has decreased. Then this type of *buy-low* investor is now using a simple feedback control strategy because gain-loss information is being used to modify the stock position over time. The ability of feedback to cope with the uncertainty of future price movements is an important advantage of its use in trading.

Closed-Loop Feedback Configuration

To describe stock trading via feedback control in a more formal manner, the first step involves the creation of a closed-loop feedback configuration involving the trader and the broker; see Fig. 1. In the figure, the feedback controller resides inside the block labeled "trader." There is a wide diversity of possible algorithms which the trader can use to modify the investment level over time. In some cases, a fixed model for future stock prices is central to the trading algorithm. Oftentimes, no stock price model is used at all, and trading signals are generated based on "price patterns." This falls under the umbrella "technical analysis" in its purest form; e.g., see the books by Kirkpatrick and Dahlquist (2007) and Lo and Hasanhodzic (2010) for further details. In any event, regardless of the trading method used, the time-varying control signal is the investment level $I(t)$.

Discrete Time and Short Selling

Since this article aims to describe real-world stock-trading mechanics as opposed to theoretical results, we work in discrete time. That is, the initial investment at time $t = 0$ is denoted by $I_0 = I(0)$, and assuming trade updates can be performed every Δt units of time, $I(t)$ is replaced by $I(k) \doteq I(k\Delta t)$. We also allow for the possibility that $I(k) < 0$. In this case, the trader is called a *short seller* and the following is meant: Shares valued at $I(k)$ are borrowed from the broker and immediately sold in the market in the hope that the price will decline. If such a decline occurs, the short seller can "cover" the position and realize a profit by buying back the stock and returning the borrowed shares to the broker. Alternatively, if the stock price increases, the short seller can continue to hold the position with a "paper loss" or buy back the borrowed stock at a loss. For the more classical case when $I(k) > 0$, the trade is said to be *long*. Finally, to conclude this section, analogous to what was done for the investment, we use the notation $p(k), g(k)$, and $V(k)$ to represent the stock price, trading gains or losses, and account value at time $t = k\Delta t$.

First Ingredient: Price Data

A trading system, be it a simulation or real-money implementation, involves sequential price data $p(k)$. This can be obtained either in real time or can be historical stock market data. As far as historical data is concerned, there are various recognized sources that provide end-of-day "closing prices," adjusted for splits and dividends. These can be downloaded for free from Yahoo! Finance. Another possibility, available from the Wharton

Stock Trading via Feedback Control, Fig. 1 Feedback loop involving trader and broker
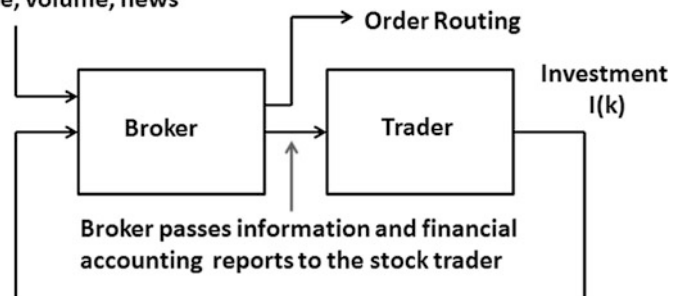
Research Data Services for a subscription fee, is the comprehensive database of historical prices at time scales from monthly to tick by tick.

It is also possible to conduct stock-trading simulations using synthetic data. For example, one of the most common ways that synthetic prices are generated is via a geometric Brownian motion process. That is, a process *drift* μ and a *volatility* $\sigma > 0$, say on an annualized basis, are provided to the simulator, and prices are generated sequentially in time via a recursion such as the Euler scheme with iterates

$$p(k+1) = \left(1 + \mu \Delta t + \sigma \epsilon(k) \sqrt{\Delta t}\right) p(k)$$

where Δt is measured in years and $\epsilon(k)$ is a zero-mean normally distributed random variable with unit standard deviation. A code used for simulation of stock trading should also include a check that $p(k) \geq 0$. The reader is referred to the textbook by Oksendal (1998) for a detailed description of this celebrated stochastic price model.

Second Ingredient: The Feedback Law

The second ingredient for trading is the previously mentioned mapping taking the information available to the trader to the amount invested $I(k)$. This feedback law is the "heart" of the controller and allows it to adapt to uncertain and changing market conditions. Perhaps the simplest example of a stock-trading *feedback law* is obtained using a classical linear time-invariant controller. In this case, the trader modulates the level of investment $I(k)$ in proportion to the cumulative gains or losses from trading according to the formula

$$I(k) = I_0 + Kg(k).$$

This is an example of technical analysis with no stock price model being used; see Fig. 2.

Using the feedback law above, the trader initially invests $I(0) = I_0$ in the stock and then begins to monitor the cumulative gain or loss $g(k)$ associated with this investment. One begins

with states $g(0) = 0$ and $I(0)$ and subsequently changes $I(k)$ if the position begins to either make or lose money depending on the movement of the stock. The constant of proportionality K above, the so-called feedback gain, is used to scale the investment level. When I_0 and K are positive, $I(k)$ is initially positive and the trade is long. Alternatively, when I_0 and K are negative, $I(k)$ is initially negative; hence, the trader is a short seller. This type of classical linear feedback is an example of a strategy which falls within the well-known class of "trend followers."

As a second example, we consider a long trade with $I_0, K > 0$ and investor who wishes to limit the trade to some level $I_{\max} > I_0$. In this case, the feedback loop includes a nonlinear saturation block, see Fig. 3, and the update equation for investment is

$$I(k) = \min\{I_0 + Kg(k), I_{\max}\}.$$

A short-trade version of the above can similarly be defined and there are also variations of this scheme, involving the notion of "reset," which assures that excessive time is not spent in the saturation regime when the stock price is falling after a long period of increase or decrease.

In the formula above and in the sequel, for simplicity, we allow $I(k)$ to represent a fractional number of shares. In practice, this type of fractional holding is only allowed in some restricted situations such as reinvestment of dividends or dollar allocations to buy shares of a mutual fund. However, in cases where a significant number of shares are being bought or sold, the use of fractional shares is a good approximation which can be used for all practical purposes. Finally, to conclude this section, we mention a subtlety which is easily overlooked in a simulation: If the intention of the trader is to be "long," then $I(k) < 0$ should be ruled out by including the condition $I(k) = \max\{I(k), 0\}$ as part of the control logic.

Order-Filling Mechanics

At time $t = k\Delta t$, the trader specifies the desired investment update to the broker who is responsible for providing a "fill" via interaction
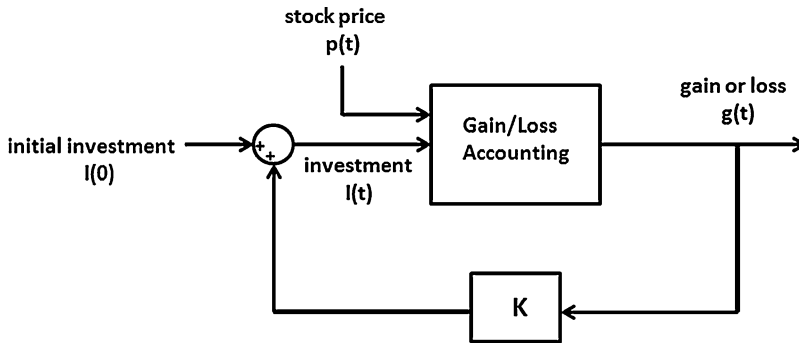
Stock Trading via Feedback Control, Fig. 2 Stock trading via linear feedback
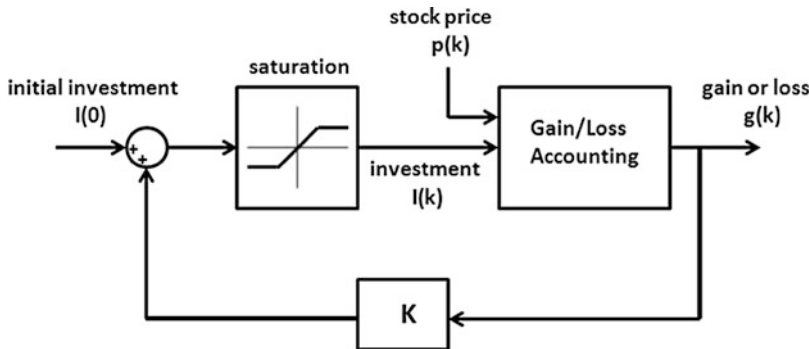
Stock Trading via Feedback Control, Fig. 3 Feedback loop with saturation

with the stock exchange. The way this step is carried out depends on a number of factors: If the stock being purchased is not heavily traded, there may be "liquidity" issues which manifest themselves as "bid-ask spread." In general, there will always exist an ask price and a bid price for any stock in the market. To see how a liquidity issue can arise, imagine a trader who wishes to purchase 100 shares at the ask price of $100 per share. If there are only 75 shares available at $100, the trader will need to pay more for the second portion of the purchase. For example, if there are 500 shares available with an ask price of $102 and transaction costs charged by the broker are 5 cents per share, the following will occur: The trader will obtain 100 shares with two "partial fills" and end up with an average acquisition cost of $100.55. This type of bid-ask gap scenario may arise for a large trader such as a hedge fund. For example, if millions of shares are being purchased at time $t = k\Delta t$, the price

of the final shares acquired may be significantly higher than the initial shares.

In the case when a stock trades with large daily volume, if large "market movers" such as hedge funds are not transacting, it can often be assumed in simulations that the trader is a *price taker*. That is, one assumes bid-ask spread is zero and trading is said to be "highly liquid." The final point to mention is that there are different order types which can be specified by the trader. The three most common *order types* are called *market*, *limit*, and *stop*.

The bottom line on order filling is as follows: When stock trading is carried out or simulated, all of the complications above can be handled via appropriate interpretation of the stock price $p(k)$ at time $t = k\Delta t$. This is accomplished as follows: When a trade is executed, be it with multiple transactions or as a special order type, we take $p(k)$ to be the average weighted price. For example, to illustrate for a long trade involving

two transactions, suppose a trader arrives at investment level $I(k)$ via two trades: the first is investment $I_a(k)$ to purchase shares at price $p_a(k)$ and the second is an investment $I_b(k)$ to purchase shares at price $p_b(k)$. Then, the average cost to acquire these shares is readily calculated to be

$$p(k) = \frac{p_a(k)p_b(k)}{p_a(k)I_b(k) + p_b(k)I_a(k)}\Delta I(k).$$

where $\Delta I(k)$ is the amount of the stock transaction at time $t = k\Delta t$. This quantity is given by

$$\Delta I(k) = I(k) - (1 + \rho(k-1))I(k-1)$$

where

$$\rho(k-1) \doteq \frac{p(k) - p(k-1)}{p(k-1)}$$

is the *percentage change* in the stock price from $k-1$ to k. Subsequently, transactions at later times $t > k\Delta t$ can be carried out as if all shares were acquired at price $p(k)$.

When this multiple-transaction issue arises in real trading, it may not be possible to predict in advance what price $p(k)$ will result. For example, in the 100-share scenario above, the outcome depended on the bid-ask queue. Notice that this did not present a problem as far as gain-loss accounting is concerned; i.e., the average price per share \$100.55 was readily calculated. However, when it comes to simulation, a model for "share acquisition" would need to be assumed. For example, for the case of geometric Brownian motion described earlier, a common model is that the trader is a price taker and that liquidity is sufficiently high so that an order involving investment $\Delta I(k)$ is filled at the sample-path price $p(k)$; i.e., no averaging over multiple transactions is required.

Gain-Loss Accounting

A broker generally provides frequent updates on gains and losses $g(k)$ attributable to stock price changes. That is,

$$g(k+1) = g(k) + \rho(k)I(k) - T(k)$$

where $T(k)$ is the so-called transaction costs, most of which consist of the broker's commission. These costs are charged for each trade and are much lower nowadays versus decades ago. For example, using a discount broker, one can easily obtain commission rates of less than \$5 per trade, even when a large number of shares are being transacted. Modulo the transaction costs, the equation above simply states that the change in the cumulative gain or loss $\Delta g(k)$ over a time increment Δt is equal to the investment $I(k)$ multiplied by the return on the stock $\Delta p(k)/p(k)$.

Interest Accumulation and Margin Charges

In many brokerage accounts, it is possible to borrow funds or shares from the broker to purchase or short sell a stock. This is referred to as trading on *margin* and the broker will charge an interest rate on the borrowed funds known as the *margin rate*. While in practice there is a limit on how much money can be borrowed, it can be quite large; e.g., hedge funds can easily obtain access to many multiples of their account value. Another possibility is that the trader is not fully invested and the account contains "idle cash" on which interest, paid by the broker, accrues.

To cover both the interest and margin accrual, we work with the *account cash*, surplus or shortfall, to determine whether interest is accrued or margin charges need to be paid. For a long trade with $I(k) > 0$ for the period Δt, we work with the *broker interest rate*, often called the risk-free return, $r_f > 0$, or the *broker margin rate m* to obtain the *interest accrual*

$$A(k) = r_f \max\{V(k) - I(k), 0\}$$
$$+ m \min\{V(k) - I(k), 0\}.$$

For the case of a short trade with $I(k) < 0$, the formula above will only hold for traders with very large accounts who have sufficient leverage with the broker so as to be allowed to capitalize on the proceeds of a short sale. For the typical

small- to medium-size trading account, the short-sale proceeds are generally "held aside" and the account is "marked to market" on a daily basis. As a result, the $A(k)$ equation above needs to be revised to account for "cash in reserve" and turns out to provide smaller interest rate accruals to the trader.

Finally, the broker's report generally includes the entire value of the account $V(k)$. This number is made up of the stock positions, either idle or borrowed cash and "dividends" $D(k)$ which may be paid periodically to the trader by the company whose shares are being held. Thus, the broker performs the calculation

$$V(k + 1) = V(0) + g(k) + A(k) + D(k)$$

and a trader can typically see these updates in real time.

Collateral Requirements and Margin Calls

When formulating the simulation model for trading, it is important to take account of the fact that the size of the trader's investment $I(k)$ is limited by the collateral requirements of the broker. For example, when a long stock position falls dramatically, a trader on margin may find that $I(k)$ exceeds the account value $V(k)$ by too large an amount to meet the broker's collateral requirements. In this case, new transactions are "stopped" and a so-called in guates results; i.e., to avoid forced liquidation of positions to bring the account back into compliance, the trader must deposit new assets or cash into the account within a short prespecified time period. In simulations, for a brokerage account with total market value $V(k)$, a constraint of the sort

$$|I(k)| \leq \gamma V(k)$$

can be imposed with $\gamma = 2$ being rather typical.

Simulation Example

We provide a simulation example illustrating the use of control in stock trading and its ability to adapt to the inherent uncertainty in stock price

movements. Figure 4 shows the daily closing prices from January 1, 2008 to June 1, 2012 of Google (GOOG), traded on the NASDAQ stock exchange. The figure also includes the 50-day simple moving average $p_{\mathrm{av}}(k)$ which will be used with a control law whose investment level depends on sign changes in $p(k) - p_{\mathrm{av}}(k)$; see Brock et al. (1992) where moving average crossing strategies are studied. There is no trading during the first 50 days while the moving average is being initialized. Subsequently, the trading begins at the first instant $k = k^*$ when the moving average has been crossed. For $k \geq k^*$, the control law for the investment level is given by

$$I(k) = I_0 \mathrm{sign}\{p(k) - p_{\mathrm{av}}(k)\}$$

where $I_0 = \$20,000$ is used in the simulation. To make the example more interesting, we assume initial account value $V(0) = \$10,000$. Hence, the issue of margin is immediately in play. In the simulation, we use risk-free rate $r_f = 0.015$ corresponding to 1.5 % per annum and a margin rate $m = 0.03$ corresponding to 3 % per annum. It is assumed that interest may be obtained on the proceeds of short sales at the risk-free rate. Google does not pay a dividend, so no adjustment of closing prices is required. A transaction cost of $3 per trade is charged. This charge occurs every day of trading because the position is adjusted daily to target $I(k) = \pm\$20,000$. We assume the broker imposes a collateral constraint of $|I(k)| \leq 2V(k)$ to limit $I(k)$ when sufficient funds are not available. Furthermore, we assume that it is possible to hold a fractional number of shares and that a "market-on-close" order each day is filled at the closing price. Finally, Fig. 4 also shows the evolution of the account value $V(k)$ over time.

Summary and Future Directions

This article concentrated entirely on trading mechanics and simulation using strategies based on control-theoretic considerations. In a future version of the encyclopedia, it would be desirable to include a "companion" article which covers the topic of *performance metrics*. That is, once trading or simulation is complete, it is natural to
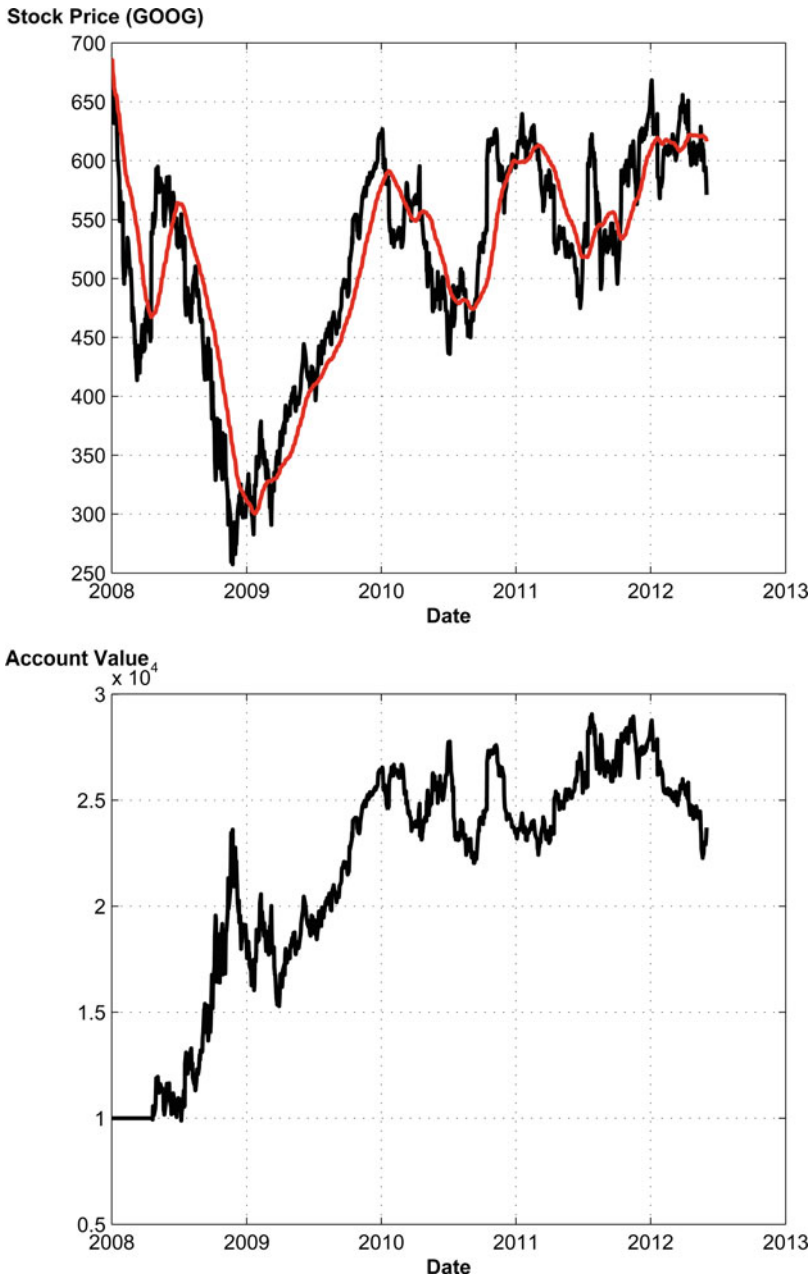
Stock Trading via Feedback Control, Fig. 4 Feedback trading of Google

ask whether the algorithm used was successful or not. To this end, there is a large body of literature covering measures for risk and return which are important for performance strategy evaluation purposes. One highlight of this literature is the paper by Artzner et al. (1999) on coherent risk measures, a topic pursued in current research.

Cross-References

- ▶ Financial Markets Modeling
- ▶ Inventory Theory
- ▶ Investment-Consumption Modeling
- ▶ Option Games: The Interface Between Optimal Stopping and Game Theory

Recommended Reading

In addition to the basic references cited in the previous sections, there is a growing body of literature on stock trading and financial markets with a control-theoretic flavor. In contrast to this article, the focal point in this literature is largely performance-related issues rather than the "nuts and bolts" of stock-trading mechanics which are described here. For the uninitiated reader, one starting reference for an overview of the literature would be the tutorial paper by Barmish et al. (2013). To provide a capsule summary, it is convenient to subdivide the literature into two categories: The first category, called *model-based* approaches, involves an underlying parameterized model structure which may or may not be completely specified. The second category of papers, called *model-free* approaches, falls under the previously mentioned umbrella of technical analysis. That is, the stock price is viewed as an external input with no predictive model for its evolution. In addition, no parameter estimation is involved and feedback trade signals are generated based on some observed "patterns" of prices or trading gains. Thus, this line of research highlights the ability of feedback to cope with the uncertainty of an unmodelled price process.

Bibliography

Artzner P, Delbaen F, Eber J, Heath D (1999) Coherent measures of risk. J Math Financ 9:203–208

Astrom KJ, Murray RM (2008) Feedback systems, an introduction for scientists and engineers. Princeton University Press, Princeton

Barmish BR, Primbs JA, Malekpour S, Warnick S (2013) On the basics for simulation of feedback-based stock trading strategies: an invited tutorial. IEEE conference on decision and control, Florence. IEEE, pp 7181–7186

Brock W, Lakonishok J, LeBaron B (1992) Simple technical trading rules and the stochastic properties of stock returns. J Financ 47:1731–1764

Kirkpatrick CD, Dahlquist JR (2007) Technical analysis: the complete resource for financial market technicians. Financial Times Press, New York

Lo AW, Hasanhodzic J (2010) The evolution of technical analysis: financial prediction from Babylonian tablets to Bloomberg terminals. Bloomberg Press, New York

Luenberger DG (1998) Investment science. Oxford, London

Merton RC (1969) Lifetime portfolio selection under uncertainty: the continuous time case. Rev Econ Stat 51:247–257

Oksendal B (1998) Stochastic differential equations: an introduction with applications. Springer, New York

Samuelson PA (1969) Lifetime portfolio selection by dynamic stochastic programming. Rev Econ Stat 51:239–246

Strategic Form Games and Nash Equilibrium

Asuman Ozdaglar
Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA

Synonyms

Nash Equilibrium

Abstract

This chapter introduces strategic form games, which provide a framework for the analysis of strategic interactions in multi-agent environments. We present the main solution concept in strategic form games, *Nash equilibrium*, and provide tools for its systematic study. We present fundamental results for existence and uniqueness of Nash equilibria and discuss their efficiency properties. We conclude with current research directions in this area.

Keywords

Efficiency; Existence; Nash equilibrium; Strategic form games; Uniqueness

Introduction

Many problems in communication, decision, and technological networks as well as in social and economic situations depend on human choices,

which are made in anticipation of the behavior of the others in the system. Examples include how to map your drive over a road network, how to use the communication medium, and how to choose strategies for resource use and more conventional economic, financial, and social decisions such as which products to buy, which technologies to invest in, or who to trust. The defining feature of all of these interactions is the dependence of an agent's objective (payoff, utility, or survival) on others' actions. Game theory focuses on formal analysis of such strategic interactions. Here, we will review strategic form games, which focus on static game-theoretic interactions and present the relevant solution concept.

Strategic Form Games

A *strategic form* game is a model for a static game in which all players act simultaneously without knowledge of other players' actions.

Definition 1 (Strategic Form Game) A strategic form game is a triplet
$\langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$ where:
1. $\mathcal{I}$ is a finite set of players, $\mathcal{I} = \{1, \ldots, I\}$.
2. S_i is a nonempty set of available actions for player i.
3. $u_i : S \to \mathbb{R}$ is the utility (payoff) function of player i where $S = \prod_{i \in \mathcal{I}} S_i$.

We will use the terms *action* and *(pure) strategy* interchangeably. (We will later use the term "mixed strategy" to refer to randomizations over actions.) We denote by $s_i \in S_i$ an action for player i, and by $s_{-i} = [s_j]_{j \neq i}$ a vector of actions for all players *except i*. We refer to the tuple $(s_i, s_{-i}) \in S$ as an *action (strategy) profile* or *outcome*. We also denote by $S_{-i} = \prod_{j \neq i} S_j$ the set of actions (strategies) of all players except i. Our convention throughout will be that each player i is interested in action profiles that "maximize" his utility function u_i.

The next two examples illustrate strategic form games with finite and infinite strategy sets.

Example 1 (Finite Strategy Sets) We consider a two-player game with finite strategy sets. Such a two-player game can be represented in matrix form, where the rows correspond to the actions of player 1 and columns represent the actions of player 2. The cell indexed by row x and column y contains a pair (a, b), where a is the payoff to player 1 and b is the payoff to player 2, i.e., $a = u_1(x, y)$ and $b = u_2(x, y)$. This class of games is sometimes referred to as *bimatrix games*. For example, consider the following game of "Matching Pennies."

|	HEADS	TAILS
HEADS	$-1, 1$	$1, -1$
TAILS	$1, -1$	$-1, 1$

Matching Pennies

This game represents "pure conflict" in the sense that one player's utility is the negative of the utility of the other player, i.e., the sum of the utilities for both players at each outcome is "zero." This class of games is referred to as *zero-sum games* (or *constant-sum games*) and has been studied extensively in the game theory literature (Basar and Olsder 1995).

Example 2 (Infinite Strategy Sets) We next present a game with infinite strategy sets. We consider a simple network game where two players send data or information flows over a communication network represented by a single link. Each player i derives a value for sending s_i units of flow over the link given by

$$v_i(s_i) = \begin{cases} a_i s_i - \frac{s_i^2}{2} & \text{if } s_i \leq a_i, \\ \frac{a_i^2}{2} & \text{if } s_i \geq a_i, \end{cases}$$

where $a_i \in [0, 1]$ is a player-specific scalar. Each player also incurs a per-flow delay or latency cost, due to congestion on the link, represented by the function $l(s) = s$, where s is the total flow on the link, i.e., $s = s_1 + s_2$ (see Fig. 1). The resulting interactions can be represented by the strategic form game $\langle \mathcal{I}, (S_i), (u_i) \rangle$, which consists of:
1. A set of two players, $\mathcal{I} = 1, 2$
2. A strategy set $S_i = [0, 1]$ for each player i, where $s_i \in S_i$ represents the amount of flow player i sends over the link
3. A utility function u_i for each player i given by value derived from sending s_i units of flow minus the total latency cost, i.e.,
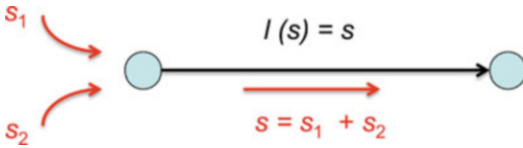
Strategic Form Games and Nash Equilibrium, Fig. 1
A network game with two players

$$u_i(s_1, s_2) = v_i(s_i) - s_i l(s_1 + s_2).$$

Nash Equilibrium

We next introduce the fundamental solution concept for strategic form games, *Nash equilibrium*. A Nash equilibrium captures a steady state of the play in a strategic form game such that each player acts optimally given their "correct" conjectures about the behavior of the other players.

Definition 2 (Nash Equilibrium) A *(pure strategy) Nash equilibrium* of a strategic form game $\langle \mathcal{I}, (S_i), (u_i)_{i \in \mathcal{I}} \rangle$ is a strategy profile $s^* \in S$ such that for all $i \in \mathcal{I}$, we have

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \qquad \text{for all } s_i \in S_i.$$

Hence, a Nash equilibrium is a strategy profile s^* such that no player i can profit by unilaterally deviating from his strategy s_i^*, assuming every other player j follows his strategy s_j^*. The definition of a Nash equilibrium can be restated in terms of best-response correspondences.

Definition 3 (Nash Equilibrium – Restated) Let $\langle \mathcal{I}, (S_i), (u_i)_{i \in \mathcal{I}} \rangle$ be a strategic form game. For any $s_{-i} \in S_{-i}$, consider the best-response correspondence of player i, $B_i(s_{-i})$, given by

$$B_i(s_{-i}) = \{s_i \in S_i \mid u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$$
$$\text{for all } s_i' \in S_i\}.$$

We say that an action profile s^* is a *Nash equilibrium* if

$$s_i^* \in B_i(s_{-i}^*) \qquad \text{for all } i \in \mathcal{I}.$$

Thus, if we define the best-response correspondence $B(s) = [B_i(s_{-i})]_{i \in \mathcal{I}}$, the set of Nash equilibria is given by the set of fixed points of $B(s)$. Below, we give two examples of games with pure strategy Nash equilibria.

Example 3 (Battle of the Sexes) Consider a two-player game with the following payoff structure:

	BALLET	SOCCER
BALLET	2, 1	0, 0
SOCCER	0, 0	1, 2

Battle of the Sexes

This game, referred to as the Battle of the Sexes game, represents a scenario in which the two players wish to coordinate their actions but have different preferences over their actions. This game has two pure strategy Nash equilibria, i.e., the strategy profiles (BALLET, BALLET) and (SOCCER, SOCCER).

Example 4 Recall the network game given in Example 2. To simplify the computations, let us assume without loss of generality that $a_1 \geq a_2 \geq \frac{a_1}{3}$. It can be seen that the best-response functions (single-valued in this case) of the players are given by

$$B_i(s_{-i}) = \max\left\{0, \frac{a_i - s_{-i}}{3}\right\} \qquad \text{for } i = 1, 2.$$

The unique pure strategy Nash equilibrium of this game is the fixed point of these functions given by

$$(s_1^*, s_2^*) = \left(\frac{3a_1 - a_2}{8}, \frac{3a_2 - a_1}{8}\right).$$

Mixed Strategy Nash Equilibrium

Consider the two-player "penalty kick" game between a penalty taker and a goalkeeper that has the same payoff structure as the matching pennies:

	LEFT	RIGHT
LEFT	1, −1	−1, 1
RIGHT	−1, 1	1, −1

Penalty kick game

This game does not have a pure strategy Nash equilibrium. It can be verified that if the penalty taker (column player) commits to a pure strategy, e.g., chooses LEFT, then the best response of the goalkeeper (row player) would be to choose the same side leading to a payoff of -1 for the penalty taker. In fact, the penalty taker would be better off following a strategy which randomizes between LEFT and RIGHT, ensuring that the goalkeeper cannot perfectly match his action. This is the idea of "randomized" or mixed strategies which we will discuss next.

We first introduce some notation. Let Σ_i denote the set of probability measures over the pure strategy (action) set S_i. We use $\sigma_i \in \Sigma_i$ to denote the *mixed strategy* of player i. When S_i is a finite set, a mixed strategy is a finite-dimensional probability vector, i.e., a vector whose elements denote the probability with which a particular action will be played. For example, if S_i has two elements, the set of mixed strategies Σ_i is the one-dimensional probability simplex, i.e., $\Sigma_i = \{(x_1, x_2) \mid x_i \geq 0, x_1 + x_2 = 1\}$. We use $\sigma \in \Sigma = \prod_{i \in \mathcal{I}} \Sigma_i$ to denote a *mixed strategy profile*. Note that this implicitly assumes that players randomize independently. We similarly denote $\sigma_{-i} \in \Sigma_{-i} = \prod_{j \neq i} \Sigma_j$.

Following von Neumann-Morgenstern expected utility theory, we extend the payoff functions u_i from S to Σ by

$$u_i(\sigma) = \int_S u_i(s) d\sigma(s),$$

i.e., the payoff of a mixed strategy σ is given by the expected value of pure strategy payoffs under the distribution σ.

We are now ready to define the mixed strategy Nash equilibrium.

Definition 4 (Mixed Strategy Nash Equilibrium) A mixed strategy profile σ^* is a *mixed strategy Nash equilibrium* if for each player i,

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \qquad \text{for all } \sigma_i \in \Sigma_i.$$

Note that since $u_i(\sigma_i, \sigma_{-i}^*) = \int_{S_i} u_i(s_i, \sigma_{-i}^*) d\sigma_i(s_i)$, it is sufficient to check only *pure* strategy

"deviations" when determining whether a given profile is a Nash equilibrium. This leads to the following characterization of a mixed strategy Nash equilibrium.

Proposition 1 *A mixed strategy profile σ^* is a mixed strategy Nash equilibrium if and only if for each player i,*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \qquad \text{for all } s_i \in S_i.$$

We also have the following useful characterization of a mixed strategy Nash equilibrium in finite strategy set games.

Proposition 2 *Let $G = \langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$ be a strategic form game with finite strategy sets. Then, $\sigma^* \in \Sigma$ is a Nash equilibrium if and only if for each player $i \in \mathcal{I}$, every pure strategy in the support of σ_i^* is a best response to σ_{-i}^*.*

Proof Let σ^* be a mixed strategy Nash equilibrium, and let $E_i^* = u_i(\sigma_i^*, \sigma_{-i}^*)$ denote the expected utility for player i. By Proposition 1, we have

$$E_i^* \geq u_i(s_i, \sigma_{-i}^*) \qquad \text{for all } s_i \in S_i.$$

We first show that $E_i^* = u_i(s_i, \sigma_{-i}^*)$ for all s_i in the support of σ_i^* (combined with the preceding relation, this proves one implication). Assume to arrive at a contradiction that this is not the case, i.e., there exists an action s_i' in the support of σ_i^* such that $u_i(s_i', \sigma_{-i}^*) < E_i^*$. Since $u_i(s_i, \sigma_{-i}^*) \leq E_i^*$ for all $s_i \in S_i$, this implies that

$$\sum_{s_i \in S_i} \sigma_i^*(s_i) u_i(s_i, \sigma_{-i}^*) < E_i^*,$$

which is a contradiction. The proof of the other implication is similar and is therefore omitted.

It follows from this characterization that every action in the support of any player's equilibrium mixed strategy yields the same payoff. This characterization extends to games with infinite strategy sets: $\sigma^* \in \Sigma$ is a Nash equilibrium if and only if for each player $i \in \mathcal{I}$, given σ_{-i}^*, no action

in S_i yields a payoff that exceeds his equilibrium payoff, and the set of actions that yields a payoff less than his equilibrium payoff has σ_i^*-measure zero.

Example 5 Let us return to the Battle of the Sexes game.

	BALLET	SOCCER
BALLET	2, 1	0, 0
SOCCER	0, 0	1, 2

Battle of the Sexes

Recall that this game has 2 pure strategy Nash equilibria. Using the characterization result in Proposition 2, we show that it has a *unique* mixed strategy Nash equilibrium (which is not a pure strategy Nash equilibrium). First, by using Proposition 2 (and inspecting the payoffs), it can be seen that there are no Nash equilibria where only one of the players randomizes over its actions. Now, assume instead that player 1 chooses the action BALLET with probability $p \in (0, 1)$ and SOCCER with probability $1 - p$ and that player 2 chooses BALLET with probability $q \in (0, 1)$ and SOCCER with probability $1 - q$. Using Proposition 2 on player 1's payoffs, we have the following relation:

$$2 \times q + 0 \times (1 - q) = 0 \times q + 1 \times (1 - q).$$

Similarly, we have

$$1 \times p + 0 \times (1 - p) = 0 \times p + 2 \times (1 - p).$$

We conclude that the only possible mixed strategy Nash equilibrium is given by $q = \frac{1}{3}$ and $p = \frac{2}{3}$.

Existence of Nash Equilibrium

The first question that one contemplates in analyzing a strategic form game is whether it has a pure or mixed strategy Nash equilibrium. While it may be possible to explicitly construct a Nash equilibrium (using either computational means or characterization results), this may be a tedious task in the case of both large finite strategy set games or infinite strategy set games with complicated utility functions. One is therefore often interested in establishing existence of an equilibrium, using conditions on the utility functions and constraint sets, before trying to understand its properties. In the sequel, we present results on existence of an equilibrium for games with finite and infinite strategy sets. The proofs of such existence results typically use fixed point arguments on the best-response correspondences of the players. They are omitted here and can be found in graduate-level game theory text books (see Fudenberg and Tirole 1991 and Myerson 1991).

Finite Strategy Set Games

We have seen that while the matching pennies game (and the penalty kick game with the same payoff structure) does not have a pure strategy Nash equilibrium, it has a mixed strategy Nash equilibrium. The next theorem, states that this existence result extends to all finite strategy set games.

Theorem 1 (Nash) *Every strategic form game with finite strategy sets has a mixed strategy Nash equilibrium.*

Infinite Strategy Set Games

A stronger result on existence of a pure strategy Nash equilibrium can be established in infinite strategy set games under some topological conditions on the utility functions and constraint sets (see Debreu 1952, Fan 1952, and Glicksberg 1952).

Theorem 2 (Debreu, Fan, Glicksberg) *Consider a strategic form game $\langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$ with infinite strategy sets such that for each $i \in \mathcal{I}$:*

1. S_i is convex and compact.
2. $u_i(s_i, s_{-i})$ is continuous in s_{-i}.
3. $u_i(s_i, s_{-i})$ is continuous and quasiconcave in s_i. (Let X be a convex set. A function $f : X \to \mathbb{R}$ is quasiconcave if every upper level set of the function, i.e., $\{x \in X \mid f(x) \geq \alpha\}$ for every scalar α, is a convex set (see Bertsekas et al. 2003).)

The game has a pure strategy Nash equilibrium.

Note that Theorem 1 is a special case of this result. For games with finite strategy sets, mixed strategy sets are simplices and hence are convex and compact, and utilities are linear in (mixed) strategies; hence, they are concave functions of (mixed) strategies (and continuous functions of mixed strategy profiles).

The next example shows that quasiconcavity cannot be dispensed with in the previous existence result.

Example 6 Consider the game where two players pick a location $s_1, s_2 \in \mathbb{R}^2$ on the circle. The payoffs are

$$u_1(s_1, s_2) = -u_2(s_1, s_2) = d(s_1, s_2),$$

where $d(s_1, s_2)$ denotes the Euclidean distance between s_1 and $s_2 \in \mathbb{R}^2$. It can be verified that this game does not have a pure strategy Nash equilibrium. However, the strategy profile where both players mix uniformly on the circle is a mixed strategy Nash equilibrium.

Without quasiconcavity, one can establish the following existence result (see Glicksberg 1952).

Theorem 3 (Glicksberg) *Consider a strategic form game $\langle \mathcal{I}, (S_i)_{i \in \mathcal{I}}, (u_i)_{i \in \mathcal{I}} \rangle$, where the S_i are nonempty compact metric spaces and the $u_i : S \to \mathbb{R}$ are continuous functions. The game has a mixed strategy Nash equilibrium.*

Uniqueness of Nash Equilibrium

Another important question that arises in the analysis of strategic form games is whether the Nash equilibrium is unique. This is important for the predictive power of Nash equilibrium since with multiple equilibria, the outcome of the game cannot be uniquely pinned down. The following result by Rosen provides sufficient conditions for uniqueness of an equilibrium in games with infinite strategy sets (see Rosen 1965). (Except for games that are strictly dominant solvable, there are no general uniqueness results for finite strategic form games.)

We first introduce some notation to state this result. Given a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$, we use the notation $\nabla f(x)$ to denote the gradient vector of f at point x, i.e.,

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \dots, \frac{\partial f(x)}{\partial x_n} \right]^T.$$

Given a scalar-valued function $F : \prod_{i=1}^{I} \mathbb{R}^{m_i} \to \mathbb{R}$, we use the notation $\nabla_i F(x)$ to denote the gradient vector of F with respect to x_i at point x, i.e.,

$$\nabla_i F(x) = \left[\frac{\partial F(x)}{\partial x_i^1}, \dots, \frac{\partial F(x)}{\partial x_i^{m_i}} \right]^T.$$

We use the notation $\nabla F(x)$ to denote

$$\nabla F(x) = [\nabla_1 F_1(x), \dots, \nabla_I F_I(x)]^T. \quad (1)$$

We assume that the strategy set S_i of each player i is given by

$$S_i = \{ x_i \in \mathbb{R}^{m_i} \mid h_i(x_i) \geq 0 \}, \quad (2)$$

where $h_i : \mathbb{R}^{m_i} \mapsto \mathbb{R}$ is a concave function. (Since h_i is concave, it follows that the set S_i is a convex set.) The next definition introduces the key condition used in establishing the uniqueness of a pure strategy Nash equilibrium.

Definition 5 We say that the utility functions $(u_1, \dots, u_I)$ are **diagonally strictly concave** for $x \in S$, if for every $x^*, \bar{x} \in S$, we have

$$(\bar{x} - x^*)^T \nabla u(x^*) + (x^* - \bar{x})^T \nabla u(\bar{x}) > 0.$$

We can now state the result on uniqueness of pure strategy Nash equilibrium in strategic form games.

Theorem 4 (Rosen) *Consider a strategic form game $\langle \mathcal{I}, (S_i), (u_i) \rangle$. For all $i \in \mathcal{I}$, assume that the strategy sets S_i are given by Eq. (2), where h_i is a concave function, and there exists some $\tilde{x}_i \in \mathbb{R}^{m_i}$ such that $h_i(\tilde{x}_i) > 0$. Assume also that the utility functions $(u_1, \dots, u_I)$ are diagonally strictly concave for $x \in S$. Then, the game has a unique pure strategy Nash equilibrium.*

S

We next provide a tractable sufficient condition for the utility functions to be diagonally strictly concave. Let $U(x)$ denote the Jacobian of $\nabla u(x)$ [see Eq. (1)]. Specifically, if the x_i are all 1-dimensional, then $U(x)$ is given by

$$U(x) = \begin{pmatrix} \frac{\partial^2 u_1(x)}{\partial x_1^2} & \frac{\partial^2 u_1(x)}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial^2 u_2(x)}{\partial x_2 \partial x_1} & \ddots & \\ \vdots & & \end{pmatrix}.$$

Proposition 3 (Rosen) *For all $i \in \mathcal{I}$, assume that the strategy sets S_i are given by Eq. (2), where h_i is a concave function. Assume that the symmetric matrix $(U(x) + U^T(x))$ is negative definite for all $x \in S$, i.e., for all $x \in S$, we have*

$$y^T (U(x) + U^T(x))y < 0, \qquad \forall \, y \neq 0.$$

Then, the payoff functions $(u_1, \ldots, u_I)$ are diagonally strictly concave for $x \in S$.

Rosen's sufficient conditions for uniqueness are quite strong. Recent work has extended such uniqueness results to hold under weaker conditions using differential topology tools. The main idea is to provide sufficient conditions so that the indices of all stationary points can be shown to be positive, which from a generalization of the Poincare-Hopf theorem (Simsek et al. 2007, 2008) implies that there exists a unique equilibrium (see Simsek et al. 2005 for applications of this methodology to several network games).

Efficiency of Nash Equilibria

Because the Nash equilibrium corresponds to the fixed point of the best-response correspondences of the players, there is no presumption that it is efficient or maximizes any well-defined weighted sum of utility functions of the players. This fact is clearly illustrated by the well-known Prisoner's Dilemma game. For some $a > 0, b > 0$, and $c > 0$ with $a > b$, the payoff matrix is given by:

	DON'T CONFESS	CONFESS
DON'T CONFESS	a, a	$b-c, a+c$
CONFESS	$a+c, b-c$	b, b

Prisoner's Dilemma

This game, generally used for capturing the dilemma of cooperation among selfish agents, has a unique (pure strategy) Nash equilibrium. (In fact each player has a dominant strategy, see Fudenberg and Tirole 1991, which is (CONFESS, CONFESS)). This clearly illustrates two aspects of the inefficiencies that arise in Nash equilibria. First, the unique Nash equilibrium is Pareto inferior meaning that if both players cooperated and chose DON'T CONFESS, they would both obtain the higher payoff of a. Second, the extent of inefficiency can be arbitrarily large based on the values of a and b. We can capture this by the *efficiency loss* (or *Price of Anarchy* as known in the literature) defined as

$$\text{Efficiency Loss} = \inf_{\text{parameters}} \frac{\sum_i u_i(\text{equilibrium})}{\sum_i u_i(\text{social optimum})},$$

where the social optimum is the strategy profile that maximizes the sum of utility functions. In the preceding example, this is clearly

$$\inf_{a,b} \frac{b}{a} = 0,$$

showing that efficiency loss can be arbitrarily large. In problems that have more structure, the efficiency loss can be bounded away from zero. A well-known example is by Pigou, which showed that in a network routing game where the congestion penalty can be described by linear latency functions (see Example 2), the efficiency loss is 3/4 (Pigou 1920). Roughgarden and Tardos in an important contribution (Roughgarden and Tardos 2000) showed that this is a lower bound for such routing games over all possible network topologies.

Summary and Future Directions

This article has provided an introduction to the basics of strategic form games. After defining the concept of Nash equilibrium, which is the basis of much of recent game theory, we have

presented fundamental results on its existence and uniqueness. We also briefly discussed issues of efficiency of Nash equilibria.

Though game theory is a mature field, there are still several important areas for inquiry. The first is a more systematic analysis and categorization of classes of games by their equilibrium and efficiency properties. Recent work by Candogan et al. (2010, 2011, 2013) provides tools for systematically analyzing equivalence classes of games that may be useful for such an investigation. The second area that is very much active concerns computational issues, which we have not considered here. Recent literature showed that computation of Nash equilibria in finite strategy set games is potentially hard and focused on developing algorithms for computing approximate Nash equilibria (see Daskalakis et al. 2006 and Lipton et al. 2003). Ongoing research in this area focuses on infinite strategy set games and exploits special structure to develop algorithms for computing (exact and approximate) Nash equilibria (Parrilo 2006; Stein et al. 2008). A third area is to develop a better application of tools of strategic form games and understand the resulting efficiency losses in networks and large-scale systems. Work in this area uses game-theoretic models to investigate resource allocation, pricing, and investment problems in networks (Johari and Tsitsiklis 2004; Acemoglu and Ozdaglar 2007; Acemoglu et al. 2009; Njoroge et al. 2013). A fourth area of research is to develop and apply alternative solution concepts for strategic form games. While some of the research in game theory has focused on subsets of Nash Equilibria (see Fudenberg and Tirole 1991), from a computational point of view, the set of correlated equilibria, which is a superset of the set of Nash Equilibria, is also attractive since it can be represented as the optimal solution set of a linear program. Correlated equilibrium can be implemented using a correlation scheme (a trusted party) or cryptographic tools as shown in Izmalkov et al. (2007). Recent work investigates alternative solution concepts for symmetric games intermediate between Nash and correlated equilibria (Stein et al. 2013), which can be implemented using specific correlation schemes.

Cross-References

▶ Dynamic Noncooperative Games
▶ Game Theory: Historical Overview
▶ Linear Quadratic Zero-Sum Two-Person Differential Games

Bibliography

Acemoglu D, Ozdaglar A (2007) Competition and efficiency in congested markets. Math Oper Res 32(1):1–31

Acemoglu D, Bimpikis K, Ozdaglar A (2009) Price and capacity competition. Games Econ Behav 66(1):1–26

Basar T, Olsder GJ (1995) Dynamic noncooperative game theory. Academic, London/New York

Bertsekas D, Nedich A, Ozdaglar A (2003) Convex analysis and optimization. Athena Scientific, Belmont

Candogan O, Ozdaglar A, Parrilo PA (2010) A projection framework for near- potential games. In: Proceedings of the IEEE conference on decision and control, CDC, Atlanta

Candogan O, Menache I, Ozdaglar A, Parrilo PA (2011) Flows and decompositions of games: harmonic and potential games. Math Oper Res 36(3):474–503

Candogan O, Ozdaglar A, Parrilo PA (2013, forthcoming) Dynamics in near-potential games. Games Econ Behav 82:66–90

Daskalakis C, Goldberg PW, Papadimitriou CH (2006) The complexity of computing a Nash equilibrium. In: Proceedings of the 38th ACM symposium on theory of computing, STOC, Seattle

Debreu D (1952) A social equilibrium existence theorem. Proc Natl Acad Sci 38:886–893

Fan K (1952) Fixed point and minimax theorems in locally convex topological linear spaces. Proc Natl Acad Sci 38:121–126

Fudenberg D, Tirole J (1991) Game theory. MIT, Cambridge

Glicksberg IL (1952) A further generalization of the Kakutani fixed point theorem with application to Nash equilibrium points. Proc Natl Acad Sci 38:170–174

Izmalkov S, Lepinski M, Micali S, Shelat A (2007) Transparent computation and correlated equilibrium. Working paper

Johari R, Tsitsiklis JN (2004) Efficiency loss in a network resource allocation game. Math Oper Res 29(3):407–435

Lipton RJ, Markakis E, Mehta A (2003) Playing large games using simple strategies. In: Proceedings of the ACM conference in electronic commerce, EC, San Diego

Myerson RB (1991) Game theory: analysis of conflict. Harvard University Press, Cambridge

Njoroge P, Ozdaglar A, Stier-Moses N, Weintraub G (2013) Investment in two-sided markets and the net

S

neutrality debate. Forthcoming in Review of Network Economics

Parrilo PA (2006) Polynomial games and sum of squares optimization. In: Proceedings of the IEEE conference on decision and control, CDC, San Diego

Pigou AC (1920) The economics of welfare. Macmillan, London

Rosen JB (1965) Existence and uniqueness of equilibrium points for concave N-person games. Econometrica 33(3):520–534

Roughgarden T, Tardos E (2000) How bad is selfish routing? In: Proceedings of the IEEE symposium on foundations of computer science, FOCS, Redondo Beach

Simsek A, Ozdaglar A, Acemoglu D (2005) Uniqueness of generalized equilibrium for box-constrained problems and applications. In: Proceedings of the Allerton conference on communication, control, and computing, Monticello

Simsek A, Ozdaglar A, Acemoglu D (2007) Generalized Poincare-Hopf theorem for compact nonsmooth regions. Math Oper Res 32(1):193–214

Simsek A, Ozdaglar A, Acemoglu D (2008) Local indices for degenerate variational inequalities. Math Oper Res 33(2):291–301

Stein N, Ozdaglar A, Parrilo PA (2008) Separable and low-rank continuous games. Int J Game Theory 37(4):475–504

Stein N, Ozdaglar A, Parrilo PA (2013) Exchangeable equilibria, part I: symmetric bimatrix games. Working paper

Stream of Variations Analysis

Jianjun Shi
Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Stream of variation (SoV) theory is a unified, model-based method for modeling, analyzing, and controlling variation in multistage manufacturing systems. A SoV model represents variation and its propagation in a multistage system using the recursive structure of state space models; such models can be derived from physical knowledge and/or estimated empirically using system operational data. Immediately, the SoV model enables integrated design and optimization for product and process tolerancing, allocation of distributed sensors in production lines, and evaluation of multistage system designs. With the help of these functions, the SoV method fulfills the objectives of system monitoring, diagnosis, and control and, ultimately, reduces a system's variation during its operation. The SoV method can be further extended to model the interactions among product quality and tooling reliability, known as the quality and reliability chain effects, which is the crucial element in carrying out quality-ensured maintenance, as well as system reliability evaluation and optimization. The SoV theory has been successfully implemented in assembly, machining, and semiconductor manufacturing processes. More research and development are needed to extend the SoV theory to manufacturing systems with complex configurations.

Keywords

Data fusion; Engineering-driven statistics; Multistage manufacturing system; Quality improvement; Variation reduction

Introduction

A multistage system refers to a system consisting of multiple units, stations, or operations to finish a final product or a service. Multistage systems are ubiquitous in modern manufacturing processes and service systems. In most cases, the final product or service quality of a multistage system is determined by complex interactions among multiple stages – the quality characteristics of one stage are not only influenced by the local variations at that stage but also by the variations propagated from upstream stages. Multistage systems present significant challenges for quality engineering research and system improvement.

The stream of variation (SoV) theory has been developed to understand and represent the complex production stream and data stream involved in the modeling and analysis of variation and its propagation in a multistage manufacturing system (Fig. 1).
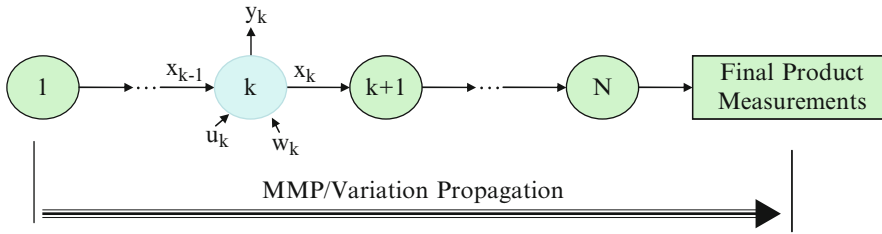
Stream of Variations Analysis, Fig. 1 Variation propagation in a multistage manufacturing process (MMP) and notations in SoV modeling (Reproduced from Shi 2006)

Stream of Variation Model

The foundation of the SoV theory is a mathematical model that links the key product quality characteristics with key process control characteristics (e.g., fixture error, machine error, etc.) in a multistage system. This model has a state space representation that describes the deviation and its propagation in an N-stage process (as shown in Fig. 1) and takes the form of

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, \quad k = 1, 2, \ldots, N, \tag{1}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \quad \{k\} \subset \{1, 2, \ldots, N\}, \tag{2}$$

where k is the stage index, $\mathbf{x}_k$ is the state vector representing the key quality characteristics of the product (or intermediate work piece) after stage k, $\mathbf{u}_k$ is the control vector representing the tooling deviations (e.g., no fault occurs if all tooling deviations are within their tolerances; fault occurs when excessive tooling deviations are beyond their tolerances; active adjustments of tooling deviations can be done to achieve error compensation objectives) at stage k, and $\mathbf{y}_k$ is the measurement vector representing product quality measurements at stage k. Vectors $\mathbf{w}_k$ and $\mathbf{v}_k$ represent modeling error and sensing error, respectively. The coefficient matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ are determined by product and process design information: $\mathbf{A}_k$ represents the impact of the deviation transition from stage $k-1$ to stage k, $\mathbf{B}_k$ represents the impact of the local tooling deviation on the product quality at stage k, and $\mathbf{C}_k$ is the measurement matrix, which can be

obtained from the defined quality features of the product at stage k.

If we repeat the modeling efforts for each stage from $k = 1$ to N, we will get the deviation and its propagation throughout the multistage manufacturing systems. By taking variances on both sides of (1) and (2) and by assuming independence among certain variables, we will obtain the variation and its propagation model for the multistage manufacturing system.

The SoV models (1) and (2) can be obtained from product and process design information and/or from the system operational data. In Shi (2006), two basic modeling methods, a physics-driven method and a data-driven method, were investigated. In the physics-driven modeling, the kinematic relationships between key control characteristics (KCC) and key product characteristics (KPC) are identified through a detailed physical analysis of the product and manufacturing process. A set of carefully defined coordinate systems are defined to represent the whole system, including the quality features in the part coordinates, part orientation to fixture/machine coordinates, and tooling to fixture/machine coordinates. Based on these coordinate systems, SoV models (1) and (2) are obtained using the state space model framework. In the data-driven modeling approach, system operational data are measured for those selected KPC and KCC variables. System identification and estimation methods are adopted to construct the SoV model. In some cases, data mining and clustering techniques are used to identify inherent relationships of the system in pre-processing. The SoV model may have different formulations, such as

the state space model, input-output model, and piecewise linear regression tree model. In most cases, engineering-driven statistical analysis is commonly used in the data analysis and modeling efforts.

With models (1) and (2), variation reduction can be achieved in both design and manufacturing phases by using mathematical optimization to make optimal decisions. However, significant challenges exist in both the model development for specific processes and model utilization to realize the benefits of the analytical capability of this model. These challenges are addressed in the SoV methodological research (Shi 2006). In more detail, the SoV methodology addresses the following important questions for variation reduction in a multistage manufacturing process.

SoV-Enabled Monitoring and Diagnosis

In multistage manufacturing systems, it is challenging to systematically find the root causes of a severe variability in terms of isolating both the manufacturing station and the underlying cause in that station. During continuous production, excessive product variation may occur at any stage of a multistage manufacturing system due to worn tooling, tooling breakage, and/or abnormal incoming part variation. The SoV theory presents systematic approaches for root cause identification. In this approach, a new concept of "statistical methods driven by engineering models" is proposed to integrate the product and process design knowledge with the on-line statistics. By solving the difference equation of models (1) and (2) and with some mathematical simplifications, the SoV model can be transformed into an input-output format as

$$\mathbf{y} = \boldsymbol{\Gamma} \cdot \mathbf{u} + \boldsymbol{\varepsilon}, \tag{3}$$

where $\mathbf{y}$ is an $n \times 1$ vector of product quality measurements, $\boldsymbol{\Gamma}$ is an $n \times p$ constant system matrix determined by product/process designs, $\mathbf{u}$ is a $p \times 1$ random vector representing the process faults, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ random vector representing measurement noises, un-modeled faults, and high-order nonlinear terms. During production, the product quality features ($\mathbf{y}$) are measured, and the data are used to conduct statistical analysis based on the model (1) to identify root causes. Two basic methods are developed for root cause diagnosis: (i) variation pattern matching: In this method, all potential variation patterns can be obtained from the matrix $\boldsymbol{\Gamma}$ resulting from the off-line system design. During the system operation, observed variation patterns can be obtained from the covariance matrix of $\mathbf{y}$. A pattern matching can be performed to identify the root causes. (ii) estimation-based diagnosis: With the SoV model and availability of on-line measurement of quality feature ($\mathbf{y}$), the deviation value of $\mathbf{u}$ can be estimated on-line. A hypothesis testing of $\mathbf{u}$ and its variance reveals the significant changes that occurred to $\mathbf{u}$, corresponding to the root causes of the system. Various estimators and their performances are evaluated in the diagnosis study (Chapter 11 of Shi 2006).

SoV-Enabled Sensor Allocation and Diagnosability

The issue of diagnosability refers to the problem of whether the product measurements contain sufficient information for the diagnosis of critical process faults, i.e., if root causes of process faults can be diagnosed. The diagnosability analysis is investigated based on model (3) that links potential process faults ($\mathbf{u}$) and product quality measurements ($\mathbf{y}$). In the SoV theory, a set of criteria is developed to evaluate the *mean* diagnosability and *variance* diagnosability for a system. Similar to observability in control theory, diagnosability is determined by the $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$ matrices ($k = 1, \ldots, N$) in the SoV models (1) and (2) (or the $\boldsymbol{\Gamma}$ matrix in model (3)). In some cases, only a subset of variables (vs. specific root cause variables) can be identified as potential root causes of the process faults, which are referred to as minimum diagnosable classes.

One emphasis in the SoV-enabled diagnosability study is to promote the concept of the "process-oriented measurement" strategy. In

current industrial practice, most of the existing measurement strategies focus on the product coherence inspection (i.e., product-oriented measurements), which is effective for detecting product imperfection, but may not be effective to identify the root causes of product quality failures. The SoV theory proposes a "process-oriented measurement" concept with a distributed sensing strategy. In this strategy, selected key control characteristics, as well as selected key product characteristics, will be measured in the selected stages for both detecting product defects and identifying their root causes.

SoV-Enabled Design and Optimization

Variation analysis and design evaluations are conducted in the product and process design stage to identify critical components, features, and manufacturing operations. With the SoV model defined in (3) and certain assumptions, we can represent the KPC-to-KCC relationship as

$$\tilde{\boldsymbol{\Sigma}}_y = \sum_{k=1}^{N} \boldsymbol{\Gamma}_k \boldsymbol{\Sigma}_{u_k} \boldsymbol{\Gamma}_k^T, \tag{4}$$

where $\tilde{\boldsymbol{\Sigma}}_y$ is the variance-covariance matrix of product quality features resulting from the variance-covariance matrix $(\boldsymbol{\Sigma}_{u_k})$ of tooling errors. Based on (3) and (4), the following four tasks can be performed: (i) tolerance analysis by allocating the tooling tolerance $(\mathbf{u}_k)$ and then predicting the final product tolerance $(\mathbf{y}_N)$; (ii) tolerance synthesis by fixing the final product tolerance $(\mathbf{y}_N)$ and then assigning the tolerance for individual tooling components $(\mathbf{u}_k)$ with certain cost objectives minimized; (iii) sensitivity study by identifying the critical tooling components $(\mathbf{u}_k)$ that have significant impacts on the final production variation through evaluation of the defined sensitivity indices; and (iv) process planning by optimizing parameters in $\mathbf{A}_k$ and $\mathbf{B}_k$ matrices to minimize the final product variation.

One unique feature of SoV-enabled design and optimization is to provide a unified method for simultaneous optimization of product and process tooling tolerance, as well as process planning. This is because the SoV models (1) and (2) represent the product quality features ($\mathbf{x}_k$ and $\mathbf{y}_k$), tooling features ($\mathbf{u}_k$), and the process planning formation ($\mathbf{A}_k$ and $\mathbf{B}_k$) within one mathematical model. As a result, a math-based optimization is feasible to achieve the best quality through process-oriented tolerance synthesis for product and process, as well as optimized process planning.

SoV-Enabled Process Control and Quality Compensation

The SoV model provides the opportunity to apply active control for dimensional variation reduction in a multistage manufacturing system. The basic idea is to implement a system-level control strategy during production to minimize the end-of-line product variance, which is propagated from upstream manufacturing stages. An optimal control scheme was devised to use the state space structure of the SoV model by treating the control as a stochastic discrete-time predictive control problem. The optimization index for determining the optimal control action is formulated as

$$J_k^* = \min_{\mathbf{u}_k} J_k = \min_{\mathbf{u}_k} E\left[\hat{\mathbf{y}}_{N|k}^T \mathbf{Q}_N \hat{\mathbf{y}}_{N|k} + \mathbf{u}_k^T \mathbf{R}_k \mathbf{u}_k\right],$$
$$\text{s.t. } C_{k,c}^{\mathrm{L}} \le u_{k,c} \le C_{k,c}^{\mathrm{U}}, \quad k = 1, \ldots, N, \; c = 1, \ldots, n_{u,k}. \tag{5}$$

where $\hat{\mathbf{y}}_{N|k}$ denotes the product quality at the final stage N that is predicted at stage k and $n_{u,k}$ is the dimension of the control action $\mathbf{u}_k$. The constraints $[\mathcal{C}_{k,c}^L, \mathcal{C}_{k,c}^U]$ define the upper and lower actuator limits that can be applied on each part/substage. $\mathbf{Q}_N \in \mathbf{R}^{m \times m}$ is a positive semi-definite matrix, and $\mathbf{R}_k \in \mathbf{R}^{n \times n}$ is a positive definite matrix.

This optimization index takes the form of the widely accepted cost function of a linear-quadratic regulator under the predictive control framework and thus satisfies the common requirements in control theory. Various research topics have been investigated under this framework, including the feed-forward control for multistage process, cautious control

considering model uncertainties, and actuator layout optimization in control system designs.

SoV-Enabled Product Quality and Reliability Chain Modeling and Analysis

There is a complex, intricate relationship between product quality and tooling reliability in a multistage manufacturing system. A degraded (or failed) production tool leads to a large variability in product quality and/or an excessive number of defects; on the other hand, excessive variability of product quality features accelerates the degradation and failure rates of production tooling at the station thereafter. For a multistage manufacturing system, these interactions are more complex as variations propagate from one stage to the next stage. Thus, a "chain effect" between the product quality (Q) and tooling reliability (R) can be observed and thus noted as the "QR chain" effect. Modeling of the QR chain is an integrated effort of the SoV model and the semi-Markov process model. The QR chain model plays an essential role in system reliability modeling and maintenance decisions and has led to new concepts of quality-ensured maintenance strategy, and tolerance synthesis considering tool degradation and system down time.

Summary and Future Directions

The concept of stream of variation for multistage systems can be applied to a very broad range of systems, although the existing work mostly focuses on the quality control of multistage discrete manufacturing processes. A comprehensive discussion on the stream of variation theory for a multistage manufacturing system is summarized in a monograph (Shi 2006). In addition, Shi and Zhou (2009) provides a survey of emerging methodologies for tackling various issues in multistage systems including modeling, analysis, monitoring, diagnosis, control, inspection, and design optimization.

The success of the multistage system framework in manufacturing processes will certainly stimulate the application of this framework to other systems. For example, monitoring and diagnosis of the abnormalities in throughput, cycle time, and lead time of a multistage production system are very promising application areas under the multistage system framework. The supply chain and logistics management, which involve multiple suppliers/venders in an interconnected fashion, can be treated as another multistage system with network structures. Most service systems such as health-care clinics, hospitals, and transportation systems are inherently multistage as well. It will be interesting to expand the stream of variation theory to these broadly defined multistage systems for their quality control, variation reduction, and other system-level performance improvement.

Cross-References

▶ Fault Detection and Diagnosis
▶ Multiscale Multivariate Statistical Process Control
▶ Statistical Process Control in Manufacturing

Recommended Reading

The monograph (Shi 2006) provides detailed results of the stream of variation theory discussed in this entry. In addition, the first five chapters of Shi (2006) provide views of basic statistical and system analysis tools needed for the SoV research and development. Some recent developments related to the SoV theory and applications are summarized in a review paper (Shi and Zhou 2009).

Bibliography

Shi J (2006) Stream of variation modeling and analysis for multistage manufacturing processes. CRC, Boca Raton, 469pp. ISBN:0-8493-2151-4

Shi J, Zhou S (2009) Quality control and improvement for multistage systems: a survey. IIE Trans Qual Reliab Eng 41:744–753

Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems

Andrew Packard[1], Peter Seiler[2], and Gary Balas[2]
[1]Mechanical Engineering Department, University of California, Berkeley, CA, USA
[2]Aerospace Engineering and Mechanics Department, University of Minnesota, Minneapolis, MN, USA

Abstract

This entry presents the most commonly used formulations of robust stability and robust $\mathcal{H}_\infty$ performance for linear systems with highly structured, linear, time-invariant uncertainty. The structured singular value function (μ) is specifically defined for this purpose, involving a problem-specific set, called the *uncertainty* set. With the uncertainty set chosen, μ is a real-valued function defined on complex matrices of a fixed dimension. A few key properties are easily derived from the definition and then applied to solve the robustness analysis problem. Computation of μ, which is required to implement the analysis tests, is difficult, so computable and refinable upper and lower bounds are derived.

Keywords

Robustness analysis; Robust control; Structured uncertainty

Notation, Definition, and Properties

$\mathbf{R}$ and $\mathbf{C}$ are the real and complex numbers; $\mathbf{C}_+ = \{\gamma \in \mathbf{C} : \text{Re}(\gamma) \geq 0\}$; $\mathbf{C}^n$ is the set of $n \times 1$ vectors and $\mathbf{C}^{n \times m}$ the set of $n \times m$ matrices

---

Gary Balas: deceased.

with elements in $\mathbf{C}$. $\bar{\sigma}(\cdot)$ refers to the maximum singular value of a matrix; for $A \in \mathbf{C}^{n \times n}$, $\rho(A)$ is the spectral radius (largest, in magnitude, eigenvalue of A), and $\rho_{\mathbf{R}}(A)$ is the real spectral radius (largest, in magnitude, real, eigenvalue of A); $\mathcal{R}$ is the ring of proper rational functions, $\mathcal{S} = \{g \in \mathcal{R} : g \text{ has no poles in } \mathbf{C}_+\}$; $\mathcal{S}^{\bullet \times \bullet}$ denotes matrices with elements in $\mathcal{S}$, where the exact dimensions are unspecified, but clear from context; finally, no notational distinction is made between a linear system, its transfer function, and/or its frequency response function.

Let R, S, and F be nonnegative integers and $r_1, \ldots, r_R$, $s_1, \ldots, s_S$, and $f_1, \ldots, f_F$ be positive integers. Define sets $\mathbf{\Delta_R} := \{\text{diag}[\delta_1 I_{r_1}, \cdots, \delta_R I_{r_R}] : \delta_i \in \mathbf{R}\}$,

$$\mathbf{\Delta_C} := \{\text{diag}[\delta_1 I_{s_1}, \cdots, \delta_S I_{s_S}, \Delta_1, \cdots, \Delta_F] \\ : \delta_i \in \mathbf{C}, \Delta_k \in \mathbf{C}^{f_k \times f_k}\}$$

and their diagonal augmentation, $\mathbf{\Delta} := \{\text{diag}[\Delta_R, \Delta_C] : \Delta_R \in \mathbf{\Delta_R}, \Delta_C \in \mathbf{\Delta_C}\} \subseteq \mathbf{C}^{n \times n}$. The set $\mathbf{\Delta}$ is called the *block structure*. The block structure can be generalized to handle nonsquare blocks in $\mathbf{\Delta_C}$ at the expense of additional notation. If $R = 0$, then $\mathbf{\Delta}$ is called a *complex* block structure. If $S = F = 0$, then $\mathbf{\Delta}$ is called a *real* block structure. For $M \in \mathbf{C}^{n \times n}$, $\mu_{\mathbf{\Delta}}(M)$ is defined as

$$\mu_{\mathbf{\Delta}}(M) := \frac{1}{\min\{\bar{\sigma}(\Delta) : \Delta \in \mathbf{\Delta}, \det(I - M\Delta) = 0\}}$$

unless no $\Delta \in \mathbf{\Delta}$ makes $I - M\Delta$ singular, in which case $\mu_{\mathbf{\Delta}}(M) := 0$, (Doyle 1982; Safonov 1982). The function $\mu_{\mathbf{\Delta}}(\cdot) : \mathbf{C}^{n \times n} \to \mathbf{R}$ is upper semicontinuous. Following Fan et al. (1991), the constraint set in the definition can be written as $\{\bar{\sigma}(\Delta) : \exists w, z \in \mathbf{C}^n, w = Mz, z = \Delta w, w \neq 0_n\}$, so that without loss of generality, at the minimum, the elements $\Delta_1, \ldots, \Delta_F$ each have rank equal to 1. For specific block structures, simplifications occur: if $R = S = 0$ and $F = 1$, then $\mu_{\mathbf{\Delta}}(M) = \bar{\sigma}(M)$; if $R = F = 0$ and $S = 1$, then $\mu_{\mathbf{\Delta}}(M) = \rho(M)$; and if $S = F = 0$ and $R = 1$, then $\mu_{\mathbf{\Delta}}(M) = \rho_{\mathbf{R}}(M)$. In general $\rho_{\mathbf{R}}(M) \leq \mu_{\mathbf{\Delta}}(M) \leq \bar{\sigma}(M)$. Associated with $\mathbf{\Delta}$ define $\mathbf{B_\Delta} := \{\Delta \in \mathbf{\Delta} : \bar{\sigma}(\Delta) \leq 1\}$. Since

S

$I - M\Delta$ is singular if and only if $M\Delta$ has an eigenvalue exactly equal to 1, it follows that $\mu_{\Delta}(M) = \max_{\Delta \in \mathbf{B}_{\Delta}} \rho_{\mathbf{R}}(M\Delta)$. If Δ is a complex block structure, then $\rho_{\mathbf{R}}(\cdot)$ can be replaced with $\rho(\cdot)$, and in that case $\mu_{\Delta}(\cdot) : \mathbf{C}^{n \times n} \to \mathbf{R}$ is continuous.

A common application is to quantify the effect (in structured singular value terms) that an uncertain matrix Δ has on the expression $F_L(M, \Delta) := M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21}$, a *linear fractional transformation* (LFT) of Δ by M. This is conceptually straightforward (informally called the *main loop theorem*) using the Schur formula for determinants. Specifically, let $\mathbf{\Delta_1} \subseteq \mathbf{C}^{n_1 \times n_1}$, $\mathbf{\Delta_2} \subseteq \mathbf{C}^{n_2 \times n_2}$ be block structures Δ and $\subseteq \mathbf{C}^{(n_1+n_2) \times (n_1+n_2)}$ be their block-diagonal augmentation. For $M \in \mathbf{C}^{(n_1+n_2) \times (n_1+n_2)}$, $\mu_{\Delta}(M) < 1$ if and only if $\mu_{\Delta_2}(M_{22}) < 1$ and

$$\max_{\Delta_2 \in \mathbf{B}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, \Delta_2)) < 1.$$

Finally (Packard and Pandey 1993) if $\mathbf{\Delta_1}$ is a block structure, and $\mathbf{\Delta_2}$ is a complex block structure, and M satisfies $\mu_{\Delta_1}(M_{11}) < \mu_{\Delta}(M)$, then $\mu_{\Delta}(\cdot)$ is continuous on an open ball around M. Loosely speaking, "if there are any complex blocks, and M is such that they matter, then μ is continuous at M." This means that at points of discontinuity, only $\Delta_R \in \mathbf{\Delta_R}$ need to be nonzero. For any polynomial $p : \mathbf{C}^n \to \mathbf{C}$, there is a minimum-norm root (using $\|\cdot\|_{\infty}$ on $\mathbf{C}^n$) whose components all have equal modulus (Doyle 1982). Defining

$$\mathbf{Q_{\Delta}} := \{\mathrm{diag}[\Delta_R, \Delta_C] : \bar{\sigma}(\Delta_R) \leq 1, \Delta_C^* \Delta_C = I\}$$

and employing this result (Young and Doyle 1997) derives that $\mu_{\Delta}(M) = \max_{Q \in \mathbf{Q_{\Delta}}} \rho_{\mathbf{R}}(MQ)$. This gives a generalized maximum-modulus-like theorem for LFTs (Packard and Pandey 1993). Revisiting the setup for the main loop theorem, assume further that $\mathbf{\Delta_2}$ is a complex block structure. If $\mu_{\Delta_2}(M_{22}) < 1$, then

$$\max_{\Delta_2 \in \mathbf{B}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, \Delta_2)) = \max_{Q_2 \in \mathbf{Q}_{\Delta_2}} \mu_{\Delta_1}(F_L(M, Q_2)).$$

This leads to specialized results per Boyd and Desoer (1985), Packard and Pandey (1993), and Tits and Fan (1995) for stable transfer function matrices. For any block structure $\mathbf{\Delta} \subseteq \mathbf{C}^{n \times n}$ and $M \in \mathcal{S}^{n \times n}$, then

$$\max\left\{\sup_{\omega \in \mathbf{R}} \mu_{\Delta}(M(j\omega)), \mu_{\Delta}(M(\infty))\right\}$$

$$= \max\left\{\sup_{s \in \mathbf{C}_+} \mu_{\Delta}(M(s)), \mu_{\Delta}(M(\infty))\right\}.$$

Robustness of Stability and Performance

There are several uncertain system formulations that all result in the same μ-analysis test to assess the robustness of stability and/or performance (Wall et al. 1982; Foo and Postlethwaite 1988). In this article, we present the simplest and most common interpretation. Consider an interconnection of known systems, $\{G_i\}_{i=1}^{M}$, and unknown systems $\{\Gamma_k\}_{k=1}^{N}$, as described by

$$\begin{bmatrix} q_1 \\ e \\ q_2 \end{bmatrix} = H \begin{bmatrix} z_1 \\ d \\ z_2 \end{bmatrix}$$

where $z_1 = \mathrm{diag}[G_1, \ldots, G_M](q_1 + w_1)$, $z_2 = \mathrm{diag}[\Gamma_1, \ldots, \Gamma_N](q_2 + w_2)$, and $H \in \mathbf{R}^{(n_1+n_e+n_2) \times (p_1+n_d+p_2)}$ (naturally partitioned as a block 3-by-3 array). This is depicted in Fig. 1. Each G_i and Γ_k is assumed to be a finite-dimensional, time-invariant linear system, with proper transfer function, and a stabilizable and detectable internal state-space description.

The interconnection is *well posed* if for any initial conditions and any (say) piecewise continuous inputs w_1, w_2, and d, there exist unique solutions to the interconnection equations. By manipulating the state-space or transfer function descriptions of a well-posed interconnection, a state-space model or proper transfer function description for the map from (d, w) to (e, z) can be derived. A well-posed interconnection is *stable* if the resultant state-space model is internally stable – the eigenvalues of its "A" matrix are in
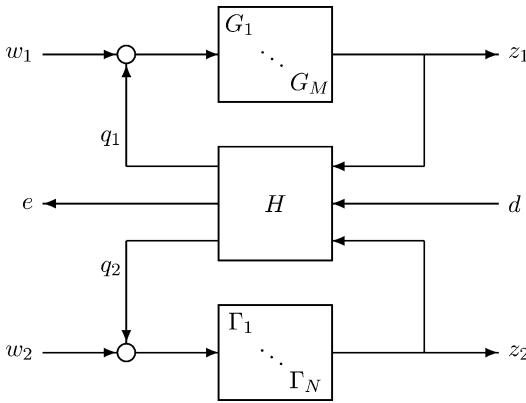
Structured Singular Value and Applications: Analyzing the Effect of Linear Time-Invariant Uncertainty in Linear Systems, Fig. 1 Interconnection of $G_1, \ldots, G_M, \Gamma_1, \ldots, \Gamma_N$

the open, left-half plane. Given some restrictions on the values of the elements of Γ, *robustness analysis* poses the question: is the interconnection well posed and stable for all possible values of Γ? And if so, then is the $\|\cdot\|_\infty$ gain from d-to-$e \le 1$ for all possible values of Γ? The goal of the analysis is to confirm "yes" or supply a particular Γ which proves that the answer is "no" (by rendering the interconnection ill-posed, unstable, or with d-to-e gain >1). Standard linear systems theory gives that the interconnection is well posed if and only if

$$\det\left(I - \begin{bmatrix} H_{11} & H_{13} \\ H_{31} & H_{33} \end{bmatrix}\begin{bmatrix} G(\infty) & 0 \\ 0 & \Gamma(\infty) \end{bmatrix}\right) \ne 0,$$

and that the interconnection is stable if and only if the transfer function matrix $T^{w,z}$, mapping $[w_1; w_2]$ to $[z_1; z_2]$, is an element of $\mathcal{S}^{\bullet \times \bullet}$.

The assumptions on each Γ_k are of three kinds: (i) Γ_k is a stable linear system, known only to satisfy $\|\Gamma_k\|_\infty < 1$; (ii) Γ_k is a stable linear system of the form $\gamma_k I$, where the scalar linear system γ_k is known to satisfy $\|\gamma_k\|_\infty < 1$; (iii) Γ_k is a constant gain, of the form $\gamma_k I$, where the scalar $\gamma_k \in \mathbf{R}$ is known to satisfy $-1 < \gamma_k < 1$. Note the similarity between this and the block structure $\boldsymbol{\Delta}$ (via $\boldsymbol{\Delta_R}$ and $\boldsymbol{\Delta_C}$) introduced earlier. After rearrangement, this block-diagonal augmentation of uncertain systems is a norm-bounded (by 1) element of the set

$$\boldsymbol{\Gamma} := \{\operatorname{diag}[\Gamma_R, \Gamma_U] : \Gamma_R \in \boldsymbol{\Delta_R}, \Gamma_U \in \mathcal{S}^{\bullet \times \bullet},$$
$$\Gamma_U(s_0) \in \boldsymbol{\Delta_C} \; \forall s_0 \in \mathbf{C_+}\}.$$

Since 0 is a possible value of Γ, two necessary conditions (denoted c.1 and c.2, respectively) for robust well-posedness and stability are at $\Gamma = 0$, specifically $\det(I - G(\infty)H_{11}) \ne 0$ and $V := G(s)(I - H_{11}G(s))^{-1} \in \mathcal{S}^{\bullet \times \bullet}$. Assuming $\det(I - G(\infty)H_{11}) \ne 0$ (i.e., c.1), the Schur formula for block determinants reduces the well-posedness condition to

$$\det\left(I - \Gamma(\infty)\left[H_{33} + H_{31}(I - G(\infty)H_{11})^{-1}\right.\right.$$
$$\left.\left. G(\infty)H_{13}\right]\right) \ne 0.$$

Define $M := H_{33} + H_{31}G(I - H_{11}G)^{-1}H_{13} \in \mathcal{S}^{\bullet \times \bullet}$, and $X := I - \Gamma M$. Then

$$T^{w,z} = \begin{bmatrix} V + VH_{13}X^{-1}\Gamma H_{31}V & VH_{13}X^{-1}\Gamma \\ X^{-1}\Gamma H_{31}V & X^{-1}\Gamma \end{bmatrix}$$

Assuming c.2, namely, $V \in \mathcal{S}^{\bullet \times \bullet}$, then $X^{-1} \in \mathcal{S}^{\bullet \times \bullet}$ implies that $T^{w,z} \in \mathcal{S}^{\bullet \times \bullet}$ – moreover $T^{w,z} \in \mathcal{S}^{\bullet \times \bullet}$ implies that $X^{-1} = I + T_{22}^{w,z}M \in \mathcal{S}^{\bullet \times \bullet}$. Finally, since both M and Γ are stable, it follows that $X^{-1} \in \mathcal{S}^{\bullet \times \bullet}$ if and only if $\det(I - M(s_0)\Gamma(s_0)) \ne 0 \quad \forall s_0 \in \mathbf{C_+}$. The maximum-modulus property gives the robustness theorem. With the definition of M and conditions c.1 and c.2, the uncertain system is robustly stable and well posed if and only if

$$\max\left\{\sup_{\omega \in \mathbf{R}} \mu_{\boldsymbol{\Delta}}(M(j\omega)), \mu_{\boldsymbol{\Delta}}(M(\infty))\right\} \le 1.$$

Indeed, if the condition holds, then by maximum-modulus theorem and the definition of μ, it follows that $\det(I - M(s)\Gamma(s)) \ne 0$ for all $s \in \mathbf{C_+}$ as well as $s = \infty$, since $\Gamma(s) \in \boldsymbol{\Delta}$ and $\bar{\sigma}(\Gamma(s)) < 1$. This gives well-posedness and stability for all such Γ, as desired (an alternate proof, using the Nyquist criterion is also common). Conversely, if the condition is violated, then at some frequency (0, nonzero, or ∞), μ is larger than 1, as evidenced by a (constant matrix) $\Delta \in \boldsymbol{\Delta} \subseteq \mathbf{C}^{n \times n}$, $\bar{\sigma}(\Delta) < 1$, which causes singularity. If the frequency is nonzero (and finite),

S

the interpolation lemmas in the appendix enable replacing the complex blocks with stable, real-rational entries. Otherwise (0 or ∞), the matrix is such that μ is continuous, and hence a finite, nonzero frequency also has $\mu > 1$, or only the real blocks are necessary to cause singularity. In all cases, $\Gamma \in \boldsymbol{\Gamma}$ with $\|\Gamma\|_{\infty} < 1$ exists to cause ill-posedness or instability (Tits and Fan 1995).

Robustness of performance, measured as $\|T^{e,d}\|_{\infty}$, can be addressed, using the main loop theorem, and an additional complex full block (recall $\bar{\sigma}(\cdot) = \mu_{\boldsymbol{\Delta}}(\cdot)$ when $F = 1$, $S = R = 0$). Define

$$M_P := \begin{bmatrix} H_{22} & H_{23} \\ H_{32} & H_{33} \end{bmatrix} + \begin{bmatrix} H_{21} \\ H_{31} \end{bmatrix}$$
$$G(I - H_{11}G)^{-1} \begin{bmatrix} H_{12} & H_{13} \end{bmatrix}$$

and $\boldsymbol{\Delta}_P := \{\operatorname{diag}[\Delta_P, \Delta] : \Delta_P \in \mathbf{C}^{n_d \times n_e}, \Delta \in \boldsymbol{\Delta}\}$. With conditions c.1 and c.2, the uncertain system is robustly stable and well posed and satisfies $\|T^{e,d}\|_{\infty} \leq 1$ if and only if

$$\max \left\{ \sup_{\omega \in \mathbf{R}} \mu_{\boldsymbol{\Delta}_P}(M_P(j\omega)), \mu_{\boldsymbol{\Delta}_P}(M_P(\infty)) \right\} \leq 1.$$

Computations

The robust stability and robust performance theorems require computing μ on the frequency response function $M(j\omega)$. Computing μ is known to be a computationally difficult problem (Toker and Ozbay 1998), so exact computational methods are generally not pursued. Reliable algorithms have been developed which yield upper and lower bounds, which are often sufficiently close for many engineering problems.

Lower Bounds

Recall that $\mu_{\boldsymbol{\Delta}}(M) = \max_{\Delta \in \mathbf{B}_{\boldsymbol{\Delta}}} \rho_{\mathbf{R}}(M\Delta) = \max_{Q \in \mathbf{Q}_{\boldsymbol{\Delta}}} \rho_{\mathbf{R}}(MQ)$. Practically speaking, these maximizations yield lower bounds for $\mu_{\boldsymbol{\Delta}}(M)$, since the global maximum may not be attained. In addition to gradient-based ascent methods, the optimality conditions for $Q \in \mathbf{Q}_{\boldsymbol{\Delta}}$ to be a local maximum of the function $\rho_{\mathbf{R}}(M\Delta)$ on the set

$\mathbf{B}_{\boldsymbol{\Delta}}$ can be derived (Young and Doyle 1997). A solution approach, similar to a Jacobi iteration, leads to an iteration that resembles combinations of the familiar power methods for spectral radius and maximum singular value. If the iteration converges (which is not guaranteed), a lower bound for $\mu_{\boldsymbol{\Delta}}(M)$ (along with a corresponding $\Delta \in \boldsymbol{\Delta}$) is produced. Studies with matrices constructed to have $\mu_{\boldsymbol{\Delta}}(M) = 1$ suggest that the iteration is very reliable for complex block structures, though usually quite poor for purely real block structures. There are several, more computationally demanding algorithms available for purely real block structures (de Gaston and Safonov 1988; Sideris and Sanchez Pena 1989). For the common situation, with both real and complex blocks, where continuity is assured, the power algorithm generally has adequate performance.

Upper Bounds

Define $\mathbf{G}_{\boldsymbol{\Delta}} := \{G = -G^* : G\Delta = -\Delta^* G^* \ \forall \Delta \in \boldsymbol{\Delta}\}$, $\mathbf{D}_{\boldsymbol{\Delta}} := \{D = D^* \succ 0 : D\Delta = \Delta D \ \forall \Delta \in \boldsymbol{\Delta}\}$, subsets of $\mathbf{C}^{n \times n}$. Elements of $\mathbf{D}_{\boldsymbol{\Delta}}$ are of the form $\operatorname{diag}[D_{r_1}, \ldots, D_{r_R}, D_{s_1}, \ldots, D_{s_S}, d_1 I_{f_1}, \ldots, d_F I_{f_F}]$, and therefore $D \in \mathbf{D}_{\boldsymbol{\Delta}}$ implies that $D^{\frac{1}{2}} \in \mathbf{D}_{\boldsymbol{\Delta}}$ too. Likewise,

$$\mathbf{G}_{\boldsymbol{\Delta}} := \{\operatorname{diag}[G_R, 0] : G_R = -G_R^* \in \mathbf{C}^{\bullet \times \bullet},$$
$$G_R \Delta_R = \Delta_R G_R \ \forall \Delta_R \in \boldsymbol{\Delta}_{\mathbf{R}}\}.$$

A concise derivation (Helmersson 1995) verifies the upper bound formula (Fan et al. 1991). If $\beta > 0$, $G \in \mathbf{G}_{\boldsymbol{\Delta}}$, and $D \in \mathbf{D}_{\boldsymbol{\Delta}}$ satisfy $M^*DM - \beta^2 D + GM + M^*G^* \preceq 0$, then $\mu_{\boldsymbol{\Delta}}(M) \leq \beta$. Indeed, if $\Delta \in \boldsymbol{\Delta}$ has $\det(I - M\Delta) = 0$, there exist nonzero $w, z \in \mathbf{C}^n$ with $w = Mz$, $z = \Delta w$. Certainly $z^*(M^*DM - \beta^2 D + GM + M^*G^*)z \leq 0$. Making substitutions gives

$$0 \geq w^*Dw - \beta^2 w^* \Delta^* D\Delta w$$
$$+ w^* \Delta^* Gw + w^* G^* \Delta w$$
$$= w^*Dw - \beta^2 w^* D^{\frac{1}{2}} \Delta^* \Delta D^{\frac{1}{2}}$$
$$w + w^* \Delta^* Gw - w^* \Delta^* Gw$$
$$= w^* D^{\frac{1}{2}} \left(I - \beta^2 \Delta^* \Delta\right) D^{\frac{1}{2}} w.$$

Since D is invertible and $w \neq 0_n$, it must be that $\bar{\sigma}(\Delta) \geq \beta^{-1}$, as desired. The constraint $M^*DM - \beta^2 D + GM + M^*G^* \preceq 0$ is a linear matrix inequality (LMI) in the variables D and G. Minimizing β over $G \in \mathbf{G_\Delta}$ and $D \in \mathbf{D_\Delta}$ subject to the LMI constraint (using Boyd and El Ghaoui 1993, for instance) yields the best upper bound that this inequality can produce.

Further Perspectives

The robustness tests involve bounding $\mu_{\mathbf{\Delta}}(M(j\omega))$ over the entire real axis. A common approach is to use a dense frequency gridding and upper/lower bound calculations at each gridded point. The advantages, simplicity and trivial parallelization, are offset with disadvantages, in that the peak value (over $\mathbf{R}$) may not be reflected accurately by the peak across the finite grid. In fact, such a grid-based test determines the smallest $\Delta \in \mathbf{\Delta}$ which can cause a pole to migrate from the left-half plane into the right-half plane *at exactly one of the frequency grid points* (as opposed to any location). Nevertheless, with some continuity assurances in place and a dense grid, this is often adequate knowledge for most engineering decisions. However, the brute-force grid approach can be avoided by treating the frequency-variable (ω) as an additional real parameter (since $M(j\omega)$ is an LFT of $\frac{1}{\omega}$) (Ferreres et al. 2003). This is a generalization of the Hamiltonian methods to compute the $\mathcal{H}_\infty$ norm of a linear system without a frequency grid, coupled with an alternative form of the upper bound (Young et al. 1995). Moreover, if only the peak value (upper bound, say) across frequency is desired, this approach can be fast, as some calculations rule out large frequency ranges to not contain the peak.

Improved upper bounds can be derived using higher-order arguments, changing the LMI constraint into a sum-of-squares constraint (which ultimately is just a larger LMI). Alternatively, branch-and-bound techniques are especially useful at reducing the conservativeness of the (D, G) upper bound when there are several real parameters ($R > 0$) (Newlin and Young 1997).

Appendix: Interpolation Lemmas

Two interpolation lemmas make the connection between robustness to constant-gain, complex-valued uncertainties ($\mathbf{\Delta}$) and stable, finite-dimensional, time-invariant linear systems described by ODEs with real coefficients ($\mathbf{\Gamma}$). Lemma 1 is used (block by block and element by element on the relevant vector directions within each block) to interpolate complex blocks causing singularity into real-rational blocks which cause singularity at a particular frequency.

Lemma 1 *Given a positive $\bar{\omega} > 0$ and a complex number δ, with $\text{Imag}(\delta) \neq 0$, there is a $\beta > 0$ such that by proper choice of sign $\pm |\delta| \left. \frac{s-\beta}{s+\beta} \right|_{s=j\bar{\omega}} = \delta$.*

Lemma 2 *Suppose $M \in \mathbf{C}^{n \times n}$ and $\bar{\omega} > 0$. If $\Delta \in \mathbf{\Delta}$ satisfies $\det(I_n - M\Delta) = 0$, then there is a $\Gamma \in \mathbf{\Gamma}$ with $\|\Gamma\|_\infty \leq \bar{\sigma}(\Delta)$ and $\det(I_n - M\Gamma(j\bar{\omega})) = 0$.*

Summary and Future Directions

The structured singular value, μ, is a linear algebra construct, defined to exactly deal with linear, time-invariant uncertainty in linear systems. The main issues are computational, focused on efficient manners to compute reasonably tight upper and lower bounds at each frequency and, more specifically, ascertain the peak value across frequency. Alternatives to the worst-case approach to robustness analysis are gaining favor and may be applicable in analysis and design situations where the abstraction of a worst-case view is too conservative (Calafiore et al. 2000).

Cross-References

▶ Fundamental Limitation of Feedback Control
▶ KYP Lemma and Generalizations/Applications
▶ Linear Systems: Continuous-Time, Time-Invariant State Variable Descriptions
▶ LMI Approach to Robust Control
▶ Optimization Based Robust Control

Recommended Reading

A comprehensive list of references, including theory, computations, and diverse applications would require many pages. The list below is minimal and does not do justice to the many researchers who have made significant contributions to this subject. In addition to the cited work, connections to Kharitonov's theorem can be found in Chen et al. (1994). Textbooks, such as Dullerud and Paganini (2000) and Zhou et al. (1996), include derivations and additional citations.

Bibliography

Boyd S, Desoer CA (1985) Subharmonic functions and performance bounds on linear time-invariant feedback systems. IMA J Math Control Inf 2:153–170

Boyd S, El Ghaoui L (1993) Method of centers for minimizing generalized eigenvalues. Linear Algebra Appl 188:63–111

Calafiore GC, Dabbene F, Tempo R (2000) Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. IEEE Trans Autom Control 45(12):2218–2235

Chen J, Fan MKH, and Nett CN (1994) Structured singular values and stability analysis of uncertain polynomials, part 1 and 2. Syst Control Lett 23:53–65 and 97–109

de Gaston RRE, Safonov MG (1988) Exact calculation of the multiloop stability margin. IEEE Trans Autom Control 33(2):156–171

Doyle J (1982) Analysis of feedback systems with structured uncertainties. IEE Proc Part D 129(6):242–250

Dullerud G, Paganini F (2000) A course in robust control theory, vol 6. Springer, New York

Fan MKH, Tits AL, Doyle JC (1991) Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. IEEE Trans Autom Control 36(1):25–38

Ferreres G, Magni JF, Biannic JM (2003) Robustness analysis of flexible structures: practical algorithms. Int J Robust Nonlinear Control 13(8):715–733

Foo YK, Postlethwaite I (1988) Extensions of the small-μ test for robust stability. IEEE Trans Autom Control 33(2):172–176

Helmersson A (1995) Methods for robust gain scheduling. PhD thesis, Linköping

Newlin MP, Young PM (1997) Mixed μ problems and branch and bound techniques. Int J Robust Nonlinear Control 7:145–164

Packard A, Pandey P (1993) Continuity properties of the real/complex structured singular value. IEEE Trans Autom Control 38(3):415–428

Safonov MG (1982) Stability margins of diagonally perturbed multivariable feedback systems. Control Theory Appl IEE Proc D 129:251–256. IET

Sideris A, Sanchez Pena RS (1989) Fast computation of the multivariable stability margin for the real interrelated uncertain parameters. IEEE Trans Autom Control 34(12):1272–1276

Tits A, Fan MKH (1995) On the small-μ theorem. Automatica 31(8):1199–1201

Toker O, Ozbay H (1998) On the complexity of purely complex mu; computation and related problems in multidimensional systems. IEEE Trans Autom Control 43(3):409–414

Wall J, Doyle JC, Stein G (1982) Performance and robustness analysis for structured uncertainty. In: IEEE conference on decision and control, Orlando, pp 629–636

Young PM, Doyle JC (1997) A lower bound for the mixed μ problem. IEEE Trans Autom Control 42(1):123–128

Young P, Newlin M, Doyle J (1995) Computing bounds for the mixed μ problem. Int J Robust Nonlinear Control 5(6):573–590

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control, vol 40. Prentice Hall, Upper Saddle River

Sub-Riemannian Optimization

Roger Brockett
Harvard University, Cambridge, MA, USA

Abstract

Optimization problems arising in the control of some important types of physical systems lead naturally to problems in sub-Riemannian optimization. Here we provide context and background material on the relevant mathematics and discuss some specific problem areas where these ideas play a role.

Keywords

Carnot-Carathéodory metric; Lie algebras; Periodic processes; Subelliptic operators; Sub-Riemannian geodesics; Symmetric spaces

Introduction

After a start in the early 1970s, over the last two decades, sub-Riemannian geometry and the related theory of subelliptic operators have become popular topics in the control literature. Their study is sometimes linked to questions involving the dynamics and control of mechanical systems with nonholonomic (nonintegrable) constraints and the use of what has classically been called quasi-coordinates because both subjects depend on Lie algebraic techniques. However, here we limit ourselves to problems in sub-Riemannian optimization per se, describing how they arise in various areas of physics and engineering. Most famously, the second law of thermodynamics, as recast by Carathéodory in differential geometric form, provides an example of the reach of sub-Riemannian geometry into the engineering world.

The statement of control theoretic problems often begins with a description of the system of interest in differential equation form:

$$\dot{x} = f(x) + \sum u_i g_i(x) \; ; \; x \in X, u \in \mathbb{R}^m$$

with X an n-dimensional manifold. In well-motivated control problems, n is almost always larger than m; the dimension of the space of controls is less than the dimension of the state space. In the case of mechanical systems, the phrase *under actuated* is sometimes used to characterize this, but the situation is ubiquitous. The analysis is complicated by presence of the immutable *drift term* f. When it is desired to use an optimization principle to find a good choice for u, one introduces a performance measure, often of the form

$$\eta = \int_0^{t_1} L(x, u) \; dt$$

and attempts to minimize η subject to whatever constraints there may be on u and x. If there is no drift term and if the Lie algebra generated by $\{g_1, g_2, \cdots, g_m\}$ defines a distribution that spans the tangent space of X at every point, the problem falls under the purview of *sub-Riemannian geometry*. In this case, one can describe the situation as $\dot{x} = G(x)u$ with G being an x-dependent rectangular matrix of rank m everywhere.

This entry is written from a control theory point of view. The problems discussed here provided the impetus for some later mathematical work, often not discussing the motivation. The purely mathematical work is de-emphasized here, much as the mathematical work often gives little or no attention to the control theoretic work that preceded it.

The Distance Function

A prototype control problem leading to sub Riemannian geometry is that of steering the system $\dot{x}_1 = u_1 \; \dot{x}_2 = u_2 \; \dot{x}_3 = x_1 u_2 - x_2 u_1$ from one state to another while minimizing

$$\eta = \int_0^1 \sqrt{u_1^2 + u_2^2} \; dt$$

It might seem that this is just a minor change from a standard shortest path problem in Riemannian geometry, e.g., it might be thought as a limiting case of a standard Riemannian geodesic problem in which the infinitesimal length is given by

$$(ds)^2 = \begin{bmatrix} dx_1 & dx_2 & dx_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ -y & x & \epsilon + x^2 + y^2 \end{bmatrix}^{-1} \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{bmatrix}$$

and ϵ is allowed to go to zero. However, because when ϵ equals zero this matrix is singular, it cannot be used to define the equations for geodesics. The most direct attack seems to be to use a Lagrange multiplier to enforce the condition on x_3, which leads to the minimization of

S

$$\eta = \int_0^1 \dot{x}_1^2 + \dot{x}_2^2 + \lambda(x_1\dot{x}_2 - x_2\dot{x}_1) \, dt$$

This yields a set of λ-dependent linear equations for x_1 and x_2. Solving these shows that the projections of the minimum length trajectories onto the (x_1, x_2)-plane are circular arcs.

In Riemannian geometry, the set of points which are of distance r from a given point will, for r sufficiently small, form a co-dimension one manifold diffeomorphic to a sphere. In this qualitative sense, Riemannian spaces are locally isotropic. In sub-Riemannian geometry, the set of points of distance $r > 0$ from a distinguished point x_0 does not have such a simple structure. For example, for the problem just discussed, we have the approximations

$$d = \sqrt{x_1^2 + x_2^2} + |x_3|/(x_1^2 + x_2^2)$$
$$\text{for } |x_3| \ll (x_1^2 + x_2^2)$$

and

$$d = 2\pi|x_3| - \sqrt{8\pi(x_1^2 + x_2^2)|x_3|}$$
$$\text{for } \sqrt{x_1^2 + x_2^2} \ll |x_3|$$

That is, for points bounded by paraboloids, defining a region near the (x_1, x_2)-plane, the distance is close to the Riemannian distance, whereas in a cone containing the x_3 axis, the distance is close to the square root of the Riemannian distance. These approximations make it clear that $d(x_1, x_2, x_3)$ is not differentiable at points on the x_3 axis. There is much more that can be said here. One interesting topic concerns the number of trajectories that satisfy the first-order necessary conditions and join a point to the origin.

More Examples

Consider the kinematic equations of the unicycle. If (x, y) are the coordinates of the center of the wheel and θ is the heading angle, then these are

$$\dot{x} = \cos\theta u_2 \; ; \; \dot{y} = \sin\theta u_2 \; ; \; \dot{\phi} = u_1$$

It is of interest to generate a "shortest path" between two points in (x, y, θ)-space where shortest is defined as the integral of some function of x, y, θ, u_1, u_2. This is typical of the kind of path planning problems in which nonholonomic constraints lead to sub-Riemannian problems. A variety of such problems arise in robotics with optimal steering programs for cars being one example.

As an example involving a compact manifold, let X be the space of 3-by-3 orthogonal matrices and consider the system described by

$$\dot{x} = \begin{bmatrix} 0 & u_1 & u_2 \\ -u_1 & 0 & 0 \\ -u_2 & 0 & 0 \end{bmatrix} x$$

In this case, the manifold X is three dimensional and the control space is two dimensional. If we wish to minimize the integral of $u^2 + v^2$ subject to $x(0) = x_0$ and $x(1) = x_1$, we have a typical sub-Riemannian geodesic problem.

If the controls contain random effects, efforts to analyze the situation lead to related problems in stochastic process. The most widely studied of these are described by an Itô equation of the form

$$dx = f(x)dt + \sum g_i(x)dw_i$$

The corresponding equation for the evolution of the probability density $\rho(t, x)$ can be put in the form

$$\frac{\partial\rho}{\partial t} = \sum a_i(x)\frac{\partial}{\partial x_i}\rho(t, x)$$
$$+ \sum b_{ij}(x)\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\rho(t, x)$$

However, rather than the right-hand side being a fully elliptic operator, as it would be in a typical heat equation (e.g., the Laplace-Beltrami operator), the symmetric matrix $B(x) = b_{ij}(x)$ is singular. If the g_i satisfy the bracket-generating condition, the density equation is said to be *subelliptic*. The system described by the Itô equation

$$\begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{bmatrix} = \begin{bmatrix} -dt & dw_1 & dw_2 \\ -dw_1 & -dt/2 & 0 \\ -dw_2 & 0 & -dt/2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

evolves on the two-sphere and the spectrum of the subelliptic operator is discrete. The diffusion time constants, i.e., the eigenvalues of the subelliptic operator, can be computed explicitly and compared with those of the fully elliptic operator, i.e., the standard Laplacian on the spherical shell.

Much has been written on the ways in which subelliptic diffusion does, and does not, share the properties of the ordinary diffusion equation.

A Special Structure

A rich, and especially tractable, class of sub-Riemannian problems come from the following situation. Suppose that $\mathcal{G}$ is a Lie group with Lie algebra G and that $\mathcal{H}$ is a closed subgroup with Lie algebra H. According to one definition, the pair $\mathcal{H} \subset \mathcal{G}$ is said to define a *symmetric space* if the Lie algebra G, viewed as a vector space, is the direct sum of H and K with $[H, K] \subset K$ and $[K, K] \subset H$. Let x evolve in $\mathcal{G}$ as

$$\dot{x} = ux \; ; \; x \in \mathcal{G} \; ; \; u \in K$$

For the sake of exposition, suppose that $\mathcal{G}$ is a matrix Lie group. We look for paths joining x_0 and x_1 that are shortest in the sense that

$$\eta = \int_0^1 ||u|| \, dt \; ;$$

is minimized, where $||u||^2 = \text{tr}(u^T u)$. (This leads to the same trajectories as those which minimize the integral of $||u||^2$.) To find the first-order necessary conditions using the maximum principle, define a Hamiltonian as $h(x, p, u) = \text{tr}(p^T ux + u^T u)$. Thus, $\dot{p} = -u^T p$ and minimizing over u implies $2u = -\pi_1(xp^T)$ where π_1 is the projection onto K. The product $m = xp^T$ satisfies $\dot{m} = [m, \pi_1(m)]$. Using the structural properties of the Lie algebra, we see that $(d/dt)\pi_0(m) = 0$ and that $(d/dt)\pi_1(m) = [\pi_1(m), m_0]$. Working out the implications, we see that trajectories of the

form $x(t) = e^{at} e^{(b-a)t}$ with $a \in H$ and $b \in K$ satisfy the first-order optimality conditions.

To illustrate, we consider the generalization of an earlier example. Let X be the space of n-by-n orthogonal matrices and consider the system described by

$$\dot{x} = \begin{bmatrix} 0 & u_1 & u_2 & \cdots & u_{n-1} \\ -u_1 & 0 & 0 & \cdots & 0 \\ -u_2 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -u_{n-1} & 0 & 0 & \cdots & 0 \end{bmatrix} x$$

Here the role of H is played by the sub-algebra of the set of real n-by-n skew-symmetric matrices consisting those whose first row and column vanish and K consists of the subset whose lower-right $(n-1)$-by-$(n-1)$ sub-matrix vanishes. In this case, the paths satisfying the first-order necessary conditions take the form $x(t) = e^{ht} e^{(k-h)t} x(0)$.

Nonintegrability and Cyclic Processes

Of course *nonintegrable* stands in opposition to the word *integrable*, as it is used in the consideration of integration performed along paths, e.g.,

$$I = \int_\gamma g_1(x)dx_1 + g_2(x)dx_2 + \cdots + g_n(x)dx_n$$

If the path γ starts at $\bar{x}$ and ends at $\hat{x}$, then the equality of mixed partials $\partial g_i/\partial x_j = \partial g_j/\partial x_i$ implies that along any two paths with these end points, the integral has the same value, provided that one of the paths can be continuously deformed into the other with the g_i being well defined along the deformation. In particular, if γ is a closed curve so $\bar{x} = \hat{x}$, then under these assumptions, the integral is zero.

On the other hand, there is a large list of important processes in biology and engineering, such as those involving the thermodynamic cycles of internal combustions engines or air conditioners, that depend critically on nonintegrable effects. These include cyclic phenomena such as

walking and breathing and a widely used mechanisms for efficient voltage conversion in electrical engineering. Thus, both nature and technology provide examples of processes in which the pistons, valves, etc. move along a smooth path and at the end of a cycle return to their initial configuration, while a related integral is not zero. Perhaps, the best-known path problem of this type is the Carnot cycle.

Questions about sub-Riemannian optimization enter here both as the optimization of the path defining the cycle and in the optimal regulation of the output of such cyclic processes. In general, the output can adjust both the amplitude and frequency of the cycle (volume of air per cycle and respiration rate), although in some cases one or the other of these might be fixed. For example, cruise control for automobiles regulates the frequency (rpm) of the engine but cannot adjust the stroke length of the pistons, whereas speed control of a running animal ordinarily involves adjusting both the length of the stride and the "steps" per minute. The primary considerations for these control processes are stability and response time, with the shape of the cycles being determined by some measure of efficiency. It seems that the optimization of such regulatory processes deserves more attention.

Cross-References

► Learning Theory
► Markov Chains and Ranking Problems in Web Search
► Modeling, Analysis, and Control with Petri Nets
► Nonlinear Adaptive Control
► Redundant Robots

Recommended Reading

Material on sub-Riemannian geometry can be found in the very readable survey (Strichartz 1986) and in more depth in Gromov (1996). The examples discussed here have mostly come from the literature Brockett (1973a,b), Baillieul (1975), and Brockett (1999) and these papers contain motivational material as well. Symmetric spaces are discussed in the sub-Riemannian context in Strichartz (1986), but for the optimization aspect, see Brockett (1999). Reference Brockett (2003) studies the regulation of sub-Riemannian cycles.

Bibliography

Baillieul J (1975) Some optimization problems in geometric control theory. PhD thesis, Harvard University

Brockett R (1973a) Lie theory and control systems defined on spheres. SIAM J Appl Math 25:213–225

Brockett R (1973b) Lie algebras and lie groups in control theory. In: Mayne DQ, Brockett RW (eds) Geometric methods in system theory. Reidel, Dordrecht, pp 43–82

Brockett R (1999) Explicitly solvable control problems with nonholonomic constraints. In: Proceedings of the 1999 CDC conference, Phoenix, pp 13–16

Brockett R (2003) Pattern generation and the control of nonlinear systems. IEEE Trans Autom Control 48:1699–1712

Gromov M (1996) Carnot-Carathéodory spaces seen from within. Sub-Riemannian geometry. Progress in mathematics, vol 144. Birkhäuser, Basel, pp 79–323

Strichartz RS (1986) Sub-Riemannian geometry. J Diff Geom 24:221–263

Subspace Techniques in System Identification

Michel Verhaegen
Delft Center for Systems and Control, Delft University, Delft, The Netherlands

Abstract

An overview is given of the class of subspace techniques (STs) for identifying linear, time-invariant state-space models from input-output data. STs do not require a parametrization of the system matrices and as a consequence do not suffer from problems related to local minima that often hamper successful application of parametric optimization- based identification methods.

The overview follows the historic line of development. It starts from Kronecker's result on the representation of an infinite power series by a rational function and then addresses, respectively, the deterministic realization problem, its stochastic variant, and finally the identification of a state-space model given in innovation form.

The overview summarizes the fundamental principles of the algorithms to solve the problems and summarizes the results about the statistical properties of the estimates as well as the practical issues like choice of weighting matrices and the selection of dimension parameters in using these STs in practice. The overview concludes with probing some future challenges and makes suggestions for further reading.

Keywords

Extended observability matrix; Hankel matrix; Innovation model; State-space model; Singular value decomposition (SVD)

Introduction

Subspace techniques (STs) for system identification address the problem of identifying state-space models of MIMO dynamical systems. The roots of ST were laid by the German mathematician Leopold Kronecker (°1823–†1891). In Kronecker (1890) Kronecker established that a power series could be represented by a rational function when the *rank of the Hankel* operator with that power series as its symbol was *finite*. In the early 1990s of the twentieth century, new generalizations of the idea of Kronecker were presented for identifying linear, time-invariant (LTI) state-space models from input-output data or output data only. These new generalizations were formulated from different perspectives, namely, within the context of canonical variate analysis (Larimore 1990), within a linear algebra context (Van Overschee and De Moor 1994; Verhaegen 1994), and subspace splitting (Jansson and Wahlberg 1996). Despite their different origin, the close relationship between these methods was quickly established by a unifying theorem that

interpreted these methods as a singular value decomposition (SVD) of a weighted matrix from which an estimate of the column space of the observability matrix or the row space of the state sequence of the given system or Kalman filter for observing the state of that system is derived (Van Overschee and De Moor 1995). This subspace calculation is the key feature that leads to the indication by ST for system identification or subspace identification methods (SIM).

The STs are attractive *complementary* techniques to the maximum likelihood or prediction error framework. They do not require the user to specify a parametrization of the system matrices of the state-space model, and the user is not confronted with the problems due to possible local minima of a nonlinear parameter optimization method that is often necessary in estimating the parameters of a state-space model via, e.g., prediction error methods. Though the statistical properties such as consistency and efficiency have been investigated, such as in Bauer and Ljung (2002), the estimates obtained via ST are in general not optimal in the statistical minimum variance sense. However, practical evidence with the use of ST in a wide variety of problems has indicated that ST provides accurate estimates. As such they are often used as an initialization to the maximum likelihood or prediction error parametric identification methods.

In this chapter we make a distinction between *output only* or stochastic identification problems and *input–output* or combined deterministic-stochastic identification problems. The first occurs when identifying, e.g., the eigenmodes of a bridge from ambient acceleration responses of the bridge. The second occurs when, in addition to ambient excitations that cannot be directly measured, controlled excitations through actuators integrated in the system are used during the collection of the input–output data.

The outline of this chapter is as follows. In the next section, we formulate the LTI state-space model identification problems and outline the general strategy of ST. The presentation of ST is given according to the historical development of ST. It starts with a summary of the solution to the deterministic realization problem, which

S

considers the noise-free "impulse" response of the system. Subsequently we present the stochastic realization problem which considers the output-only identification problem where the output is assumed to be a filtered zero-mean, white-noise sequence. The ST solution is discussed assuming samples of the covariance function of the output to be given. The deterministic-stochastic identification problem is considered in section "Combined Deterministic-Stochastic ST." In this section we first consider open-loop identification experiments. For this case, the basic linear regression problem is formulated that is at the heart of many ST. Second reference is made to a framework for analyzing and understanding the statistical properties of ST, the selection of the order, as well as to a number of open problems in the understanding of important choices the user has to made. Closed-loop identification experiments are considered in the third part of section "Combined Deterministic-Stochastic ST," while the fourth part makes a brief reference to ST papers that go beyond the LTI case.

Finally we provide a brief overview on future research directions and conclude with some recommended literature for further exploration.

ST in Identification: Problems and Strategy

The LTI system to be analyzed in this chapter is given by the following state-space model:

$$x(k + 1) = Ax(k) + Bu(k) + Ke(k)$$

$$y(k) = Cx(k) + Du(k) + e(k) \qquad (1)$$

with $u(k) \in \mathbb{R}^m$ the (measurable) input, $e(k)$ a zero-mean, white-noise sequence with $E[e(k)e(k)^T] = R$, $y(k) \in \mathbb{R}^\ell$ the (measurable) output, and $x(k) \in \mathbb{R}^n$ the state vector. This model is in the so-called innovation form since the sequence $e(k)$ is the innovation signal in a Kalman filtering context.

The historical sequence of ST developments considers the following open-loop problem formulations. In the deterministic realization problem, the innovation sequence $e(k)$ is zero, and the input $u(k)$ is an impulse. The stochastic realization problem considers the case where the input $u(k)$ is zero and the given data is assumed to be samples of the covariance function of the output. The combined deterministic-stochastic identification problem considers the model (1) for generic input $u(k)$.

The general strategy of ST is to formulate an *intermediate step* in deriving the parameters of the system matrices of interest from the given data; see Fig. 1. This intermediate step makes the ST different from the parametric model identification framework that aims for a direct estimation of the parameters of the system matrices by (in general) nonlinear parameter optimization techniques. The intermediate step in ST aims to determine a matrix from the given data that *reveals* an (approximation of an) essential subspace of the unknown system. This essential subspace can be

Intermediate ST strategy

Given Data:
$\{u(k), y(k)\}_{k=1}^N$
or $\{CA^{j-1}B\}_{j=1}^N$
etc.
— Subspace Revealing Matrix — State Space Model Realization

Direct (Non–Linear) Parameter Optimization Strategy
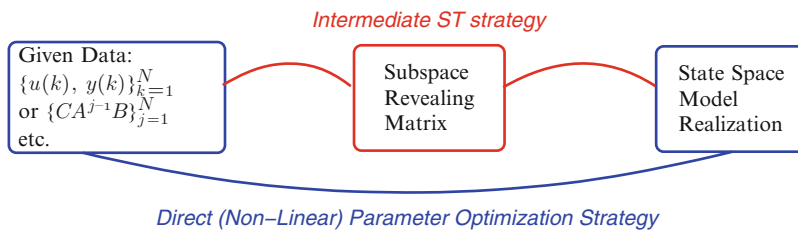
Subspace Techniques in System Identification, Fig. 1 Schematic representation of the intermediate step of ST to derive from the given data (input–output data $\{u(k), y(k)\}$, Markov parameters $\{CA^{j-1}B\}$, etc.) a subspace revealing matrix, from which the subspace of interest is computed via, e.g., singular value decomposition and that enables the computation of the state-space model realization by solving a (convex) linear least-squares problem. The commonly used approach to directly go from the given data to a state-space realization via in general nonlinear parameter optimization methods is indicated by the arrow directly connecting the given data box to the state-space realization box

the extended observability matrix of (1) as given by the matrix $\mathcal{O}_s$:

$$\mathcal{O}_s = \begin{bmatrix} C \\ CA \\ \cdots \\ CA^{s-1} \end{bmatrix} \quad \text{for} \ \ s > n,$$

or the state sequence of a Kalman filter designed for (1). Essential for ST is that both the intermediate step to reveal the subspace of interest and the subsequent derivation of the system matrices from that subspace and the given data are done via convex optimization methods and/or linear algebra methods.

Realization Theory: The Progenitor of ST

The Deterministic Realization Problem

In the 1960s, the cited result of Kronecker inspired independently Ho and Kalman, Silverman and Youla, and Tissi to present an algorithm to construct a state-space model from a Hankel matrix of impulse response coefficients (Schutter 2000). This breakthrough gave rise to the field of *realization theory*. One key problem in realization theory that paved the way for subspace identification is the determination of a minimal realization from a finite number of samples of the impulse response of a deterministic system, assumed to have a minimal representation as in (1) for $e(k) \equiv 0$. The samples of the impulse response are called the *Markov parameters*. The minimal realization sought for is the LTI model with quadruple of system matrices $[A_T, B_T, C_T, D]$, with $A_T \in \mathbb{R}^{n \times n}$ and n minimal such that the pair (A_T, C_T) is observable, the pair (A_T, B_T) is controllable, and the transfer function $D + C_T(zI - A_T)^{-1}B_T$ equals $D + C(zI - A)^{-1}B$ with z the complex variable of the z-transform. When A is stable, the latter transfer function can be written into the matrix power series:

$$D + C(zI - A)^{-1}B = D + \sum_{j=1}^{\infty} CA^{j-1}Bz^{-j} \quad (2)$$

Following the cited result of Kronecker, the solution to the minimum realization problem is based on the construction of the (block-)Hankel matrix $H_{s,N}$ constructed from the Markov parameters $\{CA^{j-1}B\}_{j=1}^{N}$ as

$$H_{s,N} = \begin{bmatrix} CB & CAB & \cdots & CA^{N-s}B \\ \vdots & & \ddots & \vdots \\ CA^{s-1}B & CA^s B & \cdots & CA^{N-1}B \end{bmatrix} \quad (3)$$

For the deterministic realization problem, the *intermediate ST step* simply is the storage of the impulse response data into a Hankel matrix. The subsequent step is to derive from this matrix a subspace from which the system matrices can be either read-off or computed via linear least squares. How this is done is outlined next.

When the order n of the minimal realization is known and the Hankel matrix dimension parameters s, N are chosen such that

$$s > n \quad N \geq 2n - 1 \quad (4)$$

the Hankel matrix $H_{s,N}$ has **rank** n. A numerically reliable way to compute that rank is via the SVD of $H_{s,N}$. Under the assumption that the rank of $H_{s,N}$ is n, we can denote that SVD as $U_n \Sigma_n V_n^T$, with $\Sigma_n \in \mathbb{R}^{n \times n}$ positive definite and with the columns of the matrices U_n and V_n orthonormal. By the minimality of (1) (for $e(k) \equiv 0$), $H_{s,N}$ can be factored as $\mathcal{O}_s \begin{bmatrix} B \ AB \cdots A^{N-s}B \end{bmatrix} = \mathcal{O}_s \mathcal{C}_{N-s+1}$ or as $\left(U_n \Sigma_n^{\frac{1}{2}} \right)\left(\Sigma_n^{\frac{1}{2}} V_n^T \right)$, and these factors are related as

$$U_n \Sigma_n^{\frac{1}{2}} = \mathcal{O}_s T^{-1} = \mathcal{O}_{s,T} \quad \Sigma_n^{\frac{1}{2}} V_n^T = T\mathcal{C}_{N-s+1}$$
$$= \mathcal{C}_{N-s+1,T}$$

for $T \in \mathbb{R}^{n \times n}$ a nonsingular transformation. Therefore $\mathcal{O}_{s,T}$ resp. $\mathcal{C}_{N-s+1,T}$ act as the extended observability resp. controllability matrix of a similarly equivalent triplet of system matrices (A_T, B_T, C_T). This correspondence allows to read-off the system matrices C_T and B_T as the first ℓ rows of the matrix $\mathcal{O}_{s,T}$ and the first m columns of $\mathcal{C}_{N-s+1,T}$ resp. Further

the *shift-invariance* property of the extended observability resp. controllability matrices allows to find the system matrix A_T of the minimal realization. For example, consider the extended observability matrix $\mathcal{O}_s$, then the shift-invariance property states that:

$$\mathcal{O}_{s,T}(1:(s-1)\ell,:)A_T = \mathcal{O}_{s,T}(\ell+1:s\ell,:) \quad (5)$$

where the notation $M(u : v, :)$ indicates the submatrix of M from rows u to rows v. The shift-invariance property delivers a set of linear equations from which the system matrix A_T can be computed via the solution of a **linear** least-squares problem when $s > n$.

Finding the dimension parameters s (and N) of the Hankel matrix $H_{s,N}$ is a nontrivial problem in general. When only the Markov parameters are given and the knowledge that they stem from a finite-order state-space model, a possible sequential strategy is to select s and N equal to the upperbounds in (4) for presumed orders n and $n + 1$, respectively. When the rank of the Hankel matrices for these two selections of s (and N) is identical, the right dimensioning of the Hankel matrix $H_{s,N}$ is found. Otherwise the presumed order is increased by one.

The Stochastic Realization Problem

The output-only identification problem aims at determining a mathematical model from a measured multivariate time series $\{y(k)\}_{k=1}^N$ with $y(k) \in \mathbb{R}^\ell$. Such a model can be then used for predicting future values of the (output) data from past values.

In the vein of the revival of the work of Kronecker on realizing dynamical systems from its impulse response, Faure and a number of contemporaries like Akaike and Aoki made pioneering contributions to extend this methodology to stochastic processes (Van Overschee and De Moor 1993). These extensions are known as solutions to the stochastic realization problem.

This problem is formulated for $y(k)$ to be a Markovian stochastic process. Reusing the notation in (1) $y(k)$ is assumed to be generated by (1) with the input $u(k) \equiv 0$. The A matrix in (1) is again assumed to be stable. The given data in

the early formulations of the stochastic realization problem was the samples of the covariance function

$$R_y(j) = E[y(k)y(k-j)^T]$$

These samples define the strictly positive real spectral density function of $y(k)$:

$$\Phi_y(z) = \sum_{j=-\infty}^{\infty} R_y(j)z^{-j} > 0 \quad (6)$$

Given the samples of the covariance function $R_y(j)$, the stochastic realization problem was to find an innovation model representation of the form

$$\hat{x}(k + 1) = A_T\hat{x}(k) + K_Te'(k)$$
$$\tilde{y}(k) = C_T\hat{x}(k) + e'(k) \quad (7)$$

with $e'(k)$ a zero-mean, white-noise input with covariance matrix R_e, the pair (A_T, C_T) observable, and A_T stable, such that the spectral density functions $\Phi_y(z)$ and $\Phi_{\tilde{y}}(z)$ are equal.

The partial similarity between this problem and the minimal realization problem becomes clear when expressing the covariance function samples $R_y(j)$ in terms of the system matrices in (1)–for $u(k) \equiv 0$ as

$$R_y(j) = CA^{j-1}G \quad \text{for} \quad j \neq 0 \quad (8)$$

with the matrices G and $R_y(0)$ derived from the following covariance expressions:

$$E[x(k)x(k)^T] = \Sigma_x : \Sigma_x$$
$$= A\Sigma_x A^T + KRK^T \quad (9)$$
$$E[x(k+1)y(k)^T] = G : G$$
$$= A\Sigma_x C^T + KR \quad (10)$$
$$E[y(k)y(k)^T] = R_y(0) : R_y(0)$$
$$= C\Sigma_x C^T + R \quad (11)$$

Since the spectral density has a two-sided series expansion, there is a so-called forward stochastic

realization problem (considering $R_y(j)$ for $j \geq 0$ only) and a backward version. Here we only treat the forward one. Drawing the parallel between the samples of the covariance function $R_y(j)$, as given in (6)–(8) and the Markov parameters in (2), we can use the deterministic tools from realization theory to find a minimal realization (A_T, C_T, G_T).

The *intermediate ST step* in the stochastic realization problem is the construction of a Hankel matrix similar to the matrix $H_{s,N}$ as in the deterministic realization problem but now from the samples of the covariance function $R_y(j)$ in (8).

With the triplet (A_T, C_T, G_T) determined, the innovation model (7) is classically completed via the solution of a Riccati equation in the unknown Σ_x. This Riccati equation results by noting that $R > 0$, and therefore, KRK^T can be written as $KR(R)^{-1}R^T K^T$. This reduces the expression for Σ_x in (9) with the help of (10) and (1) as

$$
\begin{aligned}
\Sigma_x = A\Sigma_x A^T + (G - A\Sigma_x C^T)(R_y(0) \\
- C\Sigma_x C^T)^{-1}(G - A\Sigma_x C^T)^T
\end{aligned} \quad (12)
$$

By replacing the triplet (A, C, G) with the found minimal realization (A_T, C_T, G_T) in this Riccati equation, its solution $\Sigma_{x,T}$ enables in the end to define the missing quantities as

$$
\begin{aligned}
R_e &= R_y(0) - C_T \Sigma_{x,T} C_T^T \\
K_T &= (G_T - A_T \Sigma_{x,T} C_T^T) R_e^{-1}
\end{aligned} \quad (13)
$$

By the positive realness of $\Phi_y(z)$ and the similar equivalence between the triplets (A_T, C_T, G_T) and (A, C, G), the solution $\Sigma_{x,T}$ is positive definite.

A persistent problem in solving the stochastic realization problem has existed for a long time when using approximate values of the samples $R_y(j)$. This problem is that the estimated power spectrum based on estimates of the triplet (A_T, C_T, G_T) is *no longer positive real*.

An approximate solution overcoming the problem of the loss of positive realness of the estimated power spectrum was provided in the vein of the ST developed in the early 1990s as discussed in the next section.

Combined Deterministic-Stochastic ST

Identification of LTI MIMO Systems in Open Loop

Since the golden 1960s and 1970s of the twentieth century, many attempts have been made to make the insights from deterministic and stochastic realization theory useful for system identification. To mention a few, there are attempts to use the solutions to the deterministic realization problem with measured or estimated impulse response data. One such method is known under the name of the eigensystem realization algorithm (ERA) (Juang and Pappa 1985) and has been used for modal analysis of flexible structures, like bridges, space structures, etc. Although these methods tend to work well in practice for these resonant structures that vibrate (strongly), they did not work well for other type of systems and an input different from an impulse. Extensions to the stochastic realization problem considered the use of finite sample average estimates of the covariance function as an attempt to make the method work with finite data length sequences. As indicated in section "The Stochastic Realization Problem," these approximations of the covariance function tended to violate the positive realness property of the underlying power spectrum.

In the early 1990s of the twentieth century, new breakthroughs were made working directly with the input–output data of an assumed LTI system without the need to first compute the Markov parameters or estimating the samples of covariance functions. Pioneers that contributed to these breakthroughs were Van Overschee and De Moor, introducing the N4SID approach (Van Overschee and De Moor 1994); Verhaegen, introducing the MOESP approach (Verhaegen 1994); and Larimore, presenting ST in the framework of canonical variate analysis (CVA) (Larimore 1990).

These three pioneering contributions considered the identification of the state-space model

S

(1) from the input–output data $\{u(k), y(k)\}_{k=1}^{N}$ recorded in *open loop*. The pair (A, C) was assumed to be observable, and the pair (A, KR) controllable. The innovation noise covariance matrix R was assumed to be positive definite.

The formulation of the *intermediate ST step* from which these three pioneering contributions can be derived (by weighting the result of Theorem 1) and that is at the heart of many more variants is summarized in Theorem 1. This theorem requires two preparations: first the storage of the input and output sequences into (block-) Hankel matrices and relating these Hankel matrices via the model parameters and second to make three observations about the model (1) when presented in the prediction form. This form is obtained by replacing $x(k)$ by $\hat{x}(k)$ and $e(k)$ by $y(k) - C\hat{x}(k) - Du(k)$ and is given by

$$\hat{x}(k+1) = (A - KC)\hat{x}(k)$$
$$+ (B - KD)u(k) + Ky(k)$$
$$y(k) = C\hat{x}(k) + Du(k) + e(k) \quad (14)$$

To compact the notation we make the following substitutions: $\mathcal{A} = (A - KC)$ and $\mathcal{B} = [(B - KD) \ K]$.

Let the Hankel matrix with the "future" part $\{y(k)\}_{k=p+1}^{N}$ be defined as

$$Y_f = \begin{bmatrix} y(p+1) & y(p+2) & \cdots & y(N-f+1) \\ y(p+2) & & & \\ \vdots & & \ddots & \\ y(p+f) & & \cdots & y(N) \end{bmatrix}$$
$$(15)$$

for the dimensioning parameters p and f selected such that

$$p \geq f > n$$

In a similar way we define the Hankel matrices U_f and E_f from the input $u(k)$ and the innovation $e(k)$, respectively. Then with the definition of the (block-)Toeplitz matrix T_u from the quadruple of system matrices (A, B, C, D) as

$$T_u = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & & 0 \\ CAB & CB & & 0 \\ & & \ddots & \\ CA^{f-1}B & CA^{f-2}B & \cdots & D \end{bmatrix}$$

and similarly the definition of the Toeplitz matrix T_e from the quadruple of system matrices (A, K, C, I), we can relate the data Hankel matrices Y_f and U_f as

$$Y_f = O_f \left[\hat{x}(p+1) \cdots \hat{x}(N-f+1) \right]$$
$$+ T_u U_f + T_e E_f$$
$$= O_f \hat{X}_f + T_u U_f + T_e E_f \quad (16)$$

Based on the prediction form (14), **3**, key observations are made to support the rational of the intermediate step summarized in Theorem 1:

O1: The standard assumption that the transfer function from $e(k)$ to $y(k)$ is minimum phase leads to the fact that matrix $\mathcal{A}$ is stable. Therefore, there exists a finite integer p such that
$$\mathcal{A}^p \approx 0$$

O2: The state-pace model of (14) has inputs $u(k)$ and $y(k)$. Grouping both together into the new vector $z(k) = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}$ enables to express the state $\hat{x}(k+p)$ as

$$\hat{x}(k+p) = \mathcal{A}^p \hat{x}(k) + \sum_{j=1}^{p} \mathcal{A}^{j-1} \mathcal{B} z(k+p-j)$$

for $k \geq 1$. With the assumption that $\mathcal{A}^p \approx 0$ and the definition of the input-output data vector sequence $Z(k) = \left[z(k)^T \cdots z(k+p-1)^T \right]^T$, we have the following approximation of the state:

$$\hat{x}(k+p) \approx \left[\mathcal{A}^{p-1}\mathcal{B} \cdots \mathcal{B} \right] Z(k) = \mathcal{L}^z Z(k)$$

As such the state sequence $\hat{X}_f$ in (16) can be approximated by

$$\mathcal{L}^z Z_p = \mathcal{L}^z \left[Z(1) Z(2) \cdots Z(N-f-p+1) \right].$$

O3: The (approximate) knowledge of the row space of the state sequence in $\hat{X}_f$ makes that the unknown system matrices (A, B, C, D, K) appear (approximately) linearly in the model (14).

The intermediate ST step to retrieve a matrix with relevant subspaces is summarized in the following theorem taken from Peternell et al. (1996).

Theorem 1 (Peternell et al. 1996) *Consider the model* (1) *with all stochastic processes assumed to be ergodic and with the input* $u(k)$ *to be statistically uncorrelated from the innovation* $e(\ell)$ *for all* k, ℓ. *Consider the following least-squares problem:*

$$\begin{bmatrix} \hat{L}_N^u & \hat{L}_N^z \end{bmatrix} = \arg \min_{L^u, L^z} \|Y_f - \begin{bmatrix} L^u & L^z \end{bmatrix} \begin{bmatrix} U_f \\ Z_p \end{bmatrix}\|_F^2 \tag{17}$$

with $\|.\|_F^2$ *denoting the Frobenius norm of a matrix, then*

$$\lim_{N \to \infty} \hat{L}_N^z = \mathcal{O}_f \mathcal{L}_z + \mathcal{O}_f \mathcal{A}^p \Delta_z$$

with Δ_z *a bounded matrix.*

The theorem delivers the matrix $\hat{L}_N^z$ via the solution of a convex linear least-squares problem that has asymptotically (in the number of measurements N) the extended observability matrix $\mathcal{O}_f$ as its column space and that has asymptotically (in the number of measurements as well as in the dimension parameter p) the matrix $\mathcal{L}^z$ as its row space. Based on the expression of the state sequence $\hat{X}_f$ given in the observation O2 above, the estimate of the row space of $\mathcal{L}^z$ delivers an estimate of the row space of the state sequence $\mathcal{X}_f$. The observation O3 then shows that this intermediate step allows to derive an estimate of the system matrices $[A, B, C, D, K]$ (up to a similarity transformation) via a linear least-squares problem.

Towards Understanding the Statistical Properties

Many ST variants for system identification using data recorded in open loop have been developed since the early 1990s of the twentieth century. These variants mainly differ in the use of weighting matrices $\mathcal{W}_\ell$ and $\mathcal{W}_r$ in the product $\mathcal{W}_\ell \hat{L}_N^z \mathcal{W}_r$ prior to computing the subspaces of interest. The effect on the accuracy and the statistical properties of the estimated model by these weighting matrices is yet not fully understood as is that of the dimensioning parameters p and f in the definition of the data Hankel matrices Y_f, U_f, Z_p. Only for very specific restrictions results have been achieved. For example, in Bauer and Ljung (2002), it has been shown that when the input $u(k)$ in (1) is either non-present or zero-mean white noise, as well as when the system order n of the underlying system to be known and letting in addition to the dimension parameter p and the number of data points N the dimension parameter f go to infinity, that the weighting matrices selected to represent the CVA approach (Larimore 1990) yield an optimal *minimum variance* estimate. A framework for analyzing the statistical properties like consistency and asymptotic distribution of the estimates determined by the class of STs that were discovered in the 1990s is given in Bauer (2005).

The minimum variance property of the estimates by the CVA approach (Larimore 1990) is theoretically not yet proven for more generic and practically relevant experimental conditions. For these cases, the choices of the different weighting matrices, the dimensioning parameters f, p, as well as selecting the system order are often diverted to user. Despite this fact, practical evidence has shown that STs are able to accurately identify state-space models for LTI MIMO systems under industrially realistic circumstances. As such they are by now accepted and widely used as a common engineering tool in various areas, such as model-based control, fault diagnostics, etc. Further they generally provide excellent initial estimates to the nonlinear parametric optimization methods in prediction error or maximum likelihood estimation methods.

Identification of LTI MIMO Systems in Closed Loop

The least-squares problem (17) in Theorem 1 leads to biased estimates when using

input-output data that is recorded in a closed-loop identification experiment. This is because of the correlation between the measurable input and the innovation sequence. A number of solutions have been developed to overcome this problem. We refer to the paper van der Veen et al. (2013) for an overview of a number of these rescues. A simple and performant rescue is described here based on the work in Chiuso (2010). The *intermediate ST step* in order to avoid biased estimates is to estimate a high-order vector autoregressive models with exogenous inputs, a so-called VARX model:

$$\min_{\Theta} \sum_{k=1}^{N-p} \| y(k + p) - \Theta Z(k) - Du(k + p)\|_2^2 \tag{18}$$

Using the result on the approximation of the state vector $\hat{x}(k + p)$ in observation O2, it can be shown that the solution $\hat{\Theta}$ of (18) is an approximation of the parameter vector:

$$\hat{\Theta} = \left[\widehat{C\mathcal{A}^{p-1}\mathcal{B}} \cdots \widehat{C\mathcal{B}} \right]$$

Then using this solution $\hat{\Theta}$ and O1 above leads to the following "subspace revealing matrix" (cf. Fig. 1):

$$\begin{bmatrix} \widehat{C\mathcal{A}^{p-1}\mathcal{B}} & \widehat{C\mathcal{A}^{p-2}\mathcal{B}} & \cdots & \widehat{C\mathcal{A}^{p-f}\mathcal{B}} & \cdots & \widehat{C\mathcal{B}} \\ 0 & \widehat{C\mathcal{A}^{p-1}\mathcal{B}} & \widehat{C\mathcal{A}^{p-f+1}\mathcal{B}} & \cdots & \widehat{C\mathcal{A}\mathcal{B}} \\ \vdots & & \ddots & & \\ 0 & 0 & \cdots & \widehat{C\mathcal{A}^{p-1}\mathcal{B}} & \cdots & \widehat{C\mathcal{A}^{f-1}\mathcal{B}} \end{bmatrix} \tag{19}$$

As in the open-loop case of section "Identification of LTI MIMO Systems in Open Loop," column and row weighting matrices as well as changing the size of the subspace revealing matrix (19) can be used to influence the accuracy of the estimates (Chiuso 2010). The subspace of interest of this weighted subspace revealing matrix is its row space that is an approximation of that of the state sequence $\hat{X}_f$ as in (16), now extended to make the size compatible to the weighted version of (19). Similarly as in the open-loop case, knowledge of this subspace turns the estimation of the system matrices $[A, B, C, D, K]$ (up to a similarity transformation) into a linear least-squares problem. The statistical asymptotic properties of this closed-loop ST and the treatment of the dimensioning parameters have also been studied in Chiuso (2010). Here, the result is proven that the asymptotic variance of any system invariant of the model estimated via the above closed-loop ST is a nonincreasing function of the dimensioning parameter f when the input $u(k)$ to the plant is generated by an LQG controller with a white-noise reference input.

Beyond LTI Systems

The summarized discrete-time ST methodology has been extended in various ways. A number of important extensions including representative papers are towards continuous-time systems (van der Veen et al. 2013), using frequency-domain data (Cauberghe 2006) or for different classes of nonlinear systems, like block-oriented Wiener and/or Hammerstein and linear parameter-varying systems (van Wingerden and Verhaegen 2009). ST for linear time-varying systems with changing dimension of the state vector is treated in Verhaegen and Yu (1995), and finally we mention the developments to make ST recursive (van der Veen et al. 2013).

Summary and Future Directions

Subspace techniques aim at simplifying the system identification cycle and make it more user-friendly. Still a number of challenges persist in improving on this general goal. A critical one is the "optimal" selection of the weighting matrices and the dimensioning parameters p and f of the subspace revealing matrix. Optimality here can be expressed, e.g., by the minimality of the variance of the estimates but could also be viewed more generally in relationship with the use of the model, e.g., in terms of the performance of a model-based closed-loop design. A profound theoretical framework is necessary to fully automate the selection of the weighting matrices and dimensioning and order indices. This would substantially contribute to fully automated identification procedures for doing system identification (for linear systems).

A second challenge is to better integrate ST with robust controller design. This requires the assessment of the model quality and the selection of an optimal input. Particular to the integration of ST to control design is the striking similarity of data equations used in ST and model predictive control. The challenge is to further exploit this similarity to develop data-driven model predictive control methodologies that are robust w.r.t. the identified model uncertainty.

One interesting development in ST is the use of regularization via the nuclear norm in order to improve the model order selection with respect to, e.g., SVD-based ST in Liu and Vandenberghe (2010).

A final challenge is to extend ST for LTI systems to other classes of dynamic systems, such as nonlinear, hybrid, and large-scale systems.

Cross-References

Recommended Reading

The recommended readings for further study are the books that appeared on the topic of subspace identification. In the books Verhaegen and Verdult (2007) and Katayama (2005), the topic of subspace identification is treated in a wider context for classroom teaching at the MSc level since more elaborate topics relevant in the understanding of ST are treated, such as key results from linear algebra, linear least squares, and Kalman filtering. The book Van Overschee and De Moor (1996) is focused on subspace identification only and also emphasizes the success of ST on various applications. All these books provide access to numerical implementations for getting hands-on experience with the methods. The integration of subspace methods with other identification approaches is done in the toolbox (Ljung 2007).

There also exist a number of overview articles. An overview of the early developments of ST since the 1990s of the twentieth century is given in Viberg (1995). Here also the link between ST for identifying dynamical systems and the signal processing application of direction-of-arrival problems was clearly made. A more recent overview article is van der Veen et al. (2013). In this article also reference is made to the statistical analysis and closed-loop application of ST.

Many papers have appeared reporting successful application of subspace methods in practical applications. We refer to the book Van Overschee and De Moor (1996) and the overview paper van der Veen et al. (2013).

Bibliography

Bauer D (2005) Asymptotic properties of subspace estimators. Automatica 41(3):359–376

S

Bauer D, Ljung L (2002) Some facts about the choice of the weighting matrices in larimore type of subspace algorithms. Automatica 38(5):763–773

Cauberghe B, Guillaume P, Pintelon R, Verboven P (2006) Frequency-domain subspace identification using {FRF} data from arbitrary signals. J Sound Vib 290(3–5):555–571

Chiuso A (2010) Asymptotic properties of closed-loop cca-type subspace identification. IEEE-TAC 55(3):634–649

Jansson M, Wahlberg B (1996) A linear regression approach to state-space subspace systems. Signal Process 52:103–129

Juang J-N, Pappa RS (1985) Approximate linear realizations of given dimension via Ho's algorithm. J Guid Control Dyn 8(5):620–627

Katayama T (2005) Subspace methods for system identification. Springer, London

Kronecker L (1890) Algebraische reduktion der schaaren bilinearer formen. S.B. Akad. Berlin, pp 663–776

Larimore W (1990) Canonical variate analysis in identification, filtering, and adaptive control. In: Proceedings of the 29th IEEE conference on decision and control, 1990, Honolulu, vol 2, pp 596–604

Liu Z, Vandenberghe L (2010) Interior-point method for nuclear norm approximation with application to system identification. SIAM J Matrix Anal Appl 31(3):1235–1256

Ljung L (2007) The system identification toolbox: the manual. The MathWorks Inc., Natick. 1st edition 1986, 7th edition 2007

Peternell K, Scherrer W, Deistler M (1996) Statistical analysis of novel subspace identification methods. Signal Process 52(2):161–177

Schutter BD (2000) Minimal state space realization in linear system theory: an overview. J Comput Appl Math 121(1–2):331–354

van der Veen GJ, van Wingerden JW, Bergamasco M, Lovera M, Verhaegen M (2013) Closed-loop subspace identification methods: an overview. IET Control Theory Appl 7(10):1339–1358

Van Overschee P, De Moor B (1993) Subspace algorithms for the stochastic identification problem. Automatica 29(3):649–660

Van Overschee P, De Moor B (1994) N4sid: subspace algorithms for the identification of combined deterministic-stochastic systems. Automatica 30(1):75–93

Van Overschee P, De Moor B (1995) A unifying theorem for three subspace system identification algorithms. Automatica 31(12):1853–1864

Van Overschee P, De Moor B (1996) Identification for linear systems: theory – implementation – applications. Kluwer Academic Publisher Group, Dordrecht

van Wingerden J, Verhaegen M (2009) Subspace identification of bilinear and LPV systems for open and closed loop data. Automatica 45(2):372–381

Verhaegen M (1994) Identification of the deterministic part of mimo state space models given in innovations form from input–output data. Automatica 30(1):61–74

Verhaegen M, Verdult V (2007) Filtering and identification: a least squares approach. Cambridge University Press, Cambridge/New York

Verhaegen M, Yu X (1995) A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems. Automatica 31(2):201–216

Viberg M (1995) Subspace-based methods for the identification of linear time-invariant systems. Automatica 31(12):1835–1851

Supervisory Control of Discrete-Event Systems

W.M. Wonham
Department of Electrical & Computer Engineering, University of Toronto, Toronto, ON, Canada

Abstract

We introduce background and base model for supervisory control of discrete-event systems, followed by discussion of optimal controller existence, a small example, and summary of control under partial observations. Control architecture and symbolic computation are noted as approaches to manage state space explosion.

Keywords

Asynchronous; Control architectures; Controllability; Discrete; Dynamics; Finite automata; Observability; Optimality; Regular languages; Symbolic computation

Introduction

Discrete-event (dynamic) systems (DES or DEDS) constitute a relatively new area of control science and engineering, which has taken its place in the mainstream of control research. Recently, DES have been combined with continuous systems in an area called hybrid systems.

Problems and methods for DES have been investigated for some time, although not necessarily with a "control" flavor. The parent domains can be identified as operations research and software engineering.

Operations research deals with systems of interconnected stores and servers which operate on processed items. For instance, manufacturing systems employ queues, buffers, and bins (which store workpieces). These are served by machines, robots, and automatic guided vehicles (AGVs), which process workpieces. The main problems are to measure quantitative performance and establish trade-offs, for instance flow vs. cost, and to optimize design parameters such as buffer size and maintenance frequency.

The relevant areas of software engineering include operating systems control, concurrent computing, and real-time (embedded or reactive) systems, with focus on synchronization algorithms that enforce mutual exclusion and resource sharing in the presence of concurrency, as in the classical problems of Readers & Writers and Dining Philosophers. The main objectives are (i) to guarantee safety ("Nothing bad will ever happen"), as in mutual exclusion and deadlock prevention, and (ii) to guarantee liveness ("Something good will happen eventually"), for instance, successful computational termination and eventual access to a desired resource.

DES from a Control Viewpoint

With these domains in mind, we consider DES from a control viewpoint. In general, control deals with dynamic systems, defined as entities consisting of an internal state space, together with a state-evolution or transition structure, and equipped (for control purposes) with both an input mechanism for actuation and an output channel for observation and feedback. The objective of control is to bring together information and dynamics in some purposeful combination: the interplay between observation and control or decision-making is fundamental.

In this framework, a DES is a dynamic system that is discrete, in time and usually in state

space; is asynchronous or event driven, that is driven by events or instantaneous happenings in time (which may or may not include the tick of a clock); and is nondeterministic, namely, embodies internal chance or other unmodeled mechanisms of choice which govern its state transitions. With a manufacturing system, for example, the dynamic state might include the status of machines (idle, working, down, under maintenance or repair), the contents of queues and buffers, and the locations and loads of robots and AGVs, while transitions (discrete events) occur when queues and buffers are incremented or decremented, robots load or unload, and machines start work, finish work, or break down (the "choice" between finishing work successfully and breaking down, being thus nondeterministic). In this example and many others, the objectives of design and analysis include logical correctness in the presence of concurrency and timing constraints, and quantitative performance such as rates of production, all of which depend crucially on feedback control synthesis and optimization. To this end the models will tend to be DES or hybrid systems. Nevertheless one finds the continuing relevance of standard control-theoretic concepts like feedback, stability, controllability, and observability, along with their roles in large-system architectures embodying hierarchical, decentralized, and distributed functional organization.

Here we focus on models and problems from which explicit constraints of timing are absent and which can be considered in a framework of finite-state machines and the corresponding regular languages. While the theory has been generalized to more flexible and technically advanced settings, our restricted framework is already rich enough to support numerous applications and remains challenging for large systems of industrial size.
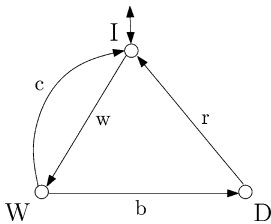
Base Model for Control of DES

The formal structure of a DES to be controlled will resemble the simple "machine" called **MACH** shown in Fig. 1. The state set of **MACH**

is $Q = \{I, W, D\}$, interpreted as Idle, Working, or Broken Down. **MACH** is initialized at state $q_o = I$, denoted by an entering arrow without source. The transition structure is displayed in Fig. 1 as a transition diagram, whose nodes are the states $q \in Q$ and edges are the transitions, each labeled with a symbol σ in the alphabet Σ, here $\{w, c, b, r\}$. If a transition (labeled) σ is an edge from q to q', then "the event σ can occur at state q." Transitions (or events) are interpreted as instantaneous in time, while states are thought of as locations where **MACH** is able to reside for some indeterminate time interval. The occurrence of w means "**MACH** enters the Working state from Idle" and similarly for c, b, r. These transitions determine the state-transition function of **MACH**, denoted by $\delta : Q \times \Sigma \to Q$. Thus $\delta(I, w) = W$, $\delta(W, b) = D$, and so on. Notice that δ is a partial function, defined at each state $q \in Q$ for only a subset of event (labels) in Σ. To denote that $\delta(q, \sigma)$ is defined at state $q \in Q$ for the event $\sigma \in \Sigma$, we write $\delta(q, \sigma)!$. The function δ can be extended in a standard way to $\delta : Q \times \Sigma^* \to Q$, where Σ^* is the set of all finite strings of elements of Σ, including the empty string ϵ. Thus $\delta(q, \epsilon) := q$ and inductively if $q' := \delta(q, s)!$, then

$$\delta(q, s.\sigma) := \delta(\delta(q, s), \sigma) := \delta(q', \sigma)$$

whenever $\delta(q', \sigma)!$. Graphically the strings $s = \sigma_1 \ldots \sigma_k \in \Sigma^*$ for which $\delta(q, s)!$ are precisely those for which there exists a path in the transition diagram starting from q and having successive edges labeled $\sigma_1, \ldots, \sigma_k$.

We call any subset of Σ^* (i.e., any set of strings of elements from Σ) a language over Σ and accordingly speak of sublanguages of a language over Σ.

For **MACH**, the execution of a production cycle, namely the event sequence (or string) $w.c$, or a work-breakdown-repair cycle, the string $w.b.r$, can be considered successful, and the corresponding string is said to be marked. States which are entered by marked strings are marked states and identified in a transition diagram by an outgoing arrow with no target. In Fig. 1, the only marked state happens to be the initial state, which is thus shown with a double arrow; in general there could be several marked states, which may or may not include the initial state. The marked states comprise a subset $Q_m \subseteq Q$, which may be empty (at one extreme) or equal to Q (at the other). The case $Q_m = Q$ (all states marked) would imply that every string of events is considered as significant or successful as any other, while the case $Q_m = \emptyset$ (no state marked, so there are no successful strings) plays a technical role in computation.

In general a generator is a tuple $\mathbf{G} = (Q, \Sigma, \delta, q_o, Q_m)$ usually interpreted physically as for **MACH** above, but mathematically consisting merely of the finite-state set Q, finite alphabet Σ, marked subset $Q_m \subseteq Q$, with initial state $q_o \in Q$, and (partial) transition function $\delta : Q \times \Sigma \to Q$. Additionally we bring in the closed behavior $L(\mathbf{G})$ of $\mathbf{G}$, defined as all the strings of Σ^* which $\mathbf{G}$ can generate starting from the initial state, in the sense

$$L(\mathbf{G}) := \{s \in \Sigma^* \mid \delta(q_o, s)!\}.$$

Of central importance also is the marked behavior of $\mathbf{G}$, namely, the sublanguage of $L(\mathbf{G})$ given by

$$L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) \mid \delta(q_o, s) \in Q_m\}.$$

We need several definitions. A string s' is a prefix of a string $s \in \Sigma^*$, written $s' \leq s$, if s' can be extended to s, namely, there exists a string w in Σ^* such that $s'.w = s$. The closure of a language $M \subseteq \Sigma^*$ is the language $\overline{M}$ consisting of all prefixes of strings in M:

$$\overline{M} := \{s' \in \Sigma^* \mid s' \leq s \text{ for some } s \text{ in } M\}$$

Supervisory Control of Discrete-Event Systems, Fig. 1
MACH

A language N over Σ is (prefix-)closed if it contains all its prefixes, namely, $N = \overline{N}$. In this notation $\mathbf{G}$ is said to be nonblocking if $L(\mathbf{G}) = \overline{L_m(\mathbf{G})}$, namely, any (generated) string in $L(\mathbf{G})$ is a prefix of, and so can be extended to, a marked string of $\mathbf{G}$.

The semantics of $\mathbf{G}$ (its mathematical meaning) is simply the pair of languages $L_m(\mathbf{G})$, $L(\mathbf{G})$. In general the latter may be infinite subsets of Σ^*, while $\mathbf{G}$ itself is a finite object, considered to represent an algorithm for the generation of its behaviors. Unless $\mathbf{G}$ is trivial (has empty state set), it is always true that $\epsilon \in L(\mathbf{G})$.

Transition labeling of $\mathbf{G}$ is deterministic: at every q, at most one transition is defined for each given event σ, namely,

$$\delta(q, \sigma) = q' \ \& \ \delta(q, \sigma) = q'' \text{ implies } q' = q''.$$

It is quite acceptable, however, that at distinct states q and r, both $\delta(q, \sigma)!$ and $\delta(r, \sigma)!$ (where these evaluations are usually not equal).

To formulate a control problem for $\mathbf{G}$, we first adjoin a control technology or mechanism by which $\mathbf{G}$ may be actuated to affect its temporal behavior, namely, determine the strings it is permitted to generate. To this end we assume that a subset of events $\Sigma_c \subseteq \Sigma$, called the controllable events, are capable of being enabled or disabled by an external controller. Think of a traffic light being turned green or red to allow or prohibit passage (vehicle transition) through an intersection. The complementary event subset $\Sigma_u := \Sigma - \Sigma_c$ is uncontrollable; events $\sigma \in \Sigma_u$ cannot be externally disabled but may be considered permanently enabled. For $\mathbf{G} = \mathbf{MACH}$ one might reasonably assume $\Sigma_c = \{w, r\}$, $\Sigma_u = \{c, b\}$. At a given state q of $\mathbf{G}$, it will be true in general that $\delta(q, \sigma)!$ both for some (controllable) events $\sigma \in \Sigma_c$ and for some (uncontrollable) events $\sigma \in \Sigma_u$. Among the $\sigma \in \Sigma_c$, at a given time, some may be externally enabled and others disabled. So, $\mathbf{G}$ will nondeterministically choose its next generated event from the subset

$$\{\sigma \in \Sigma_u \mid \delta(q, \sigma)!\} \cup \{\sigma \in \Sigma_c \mid \delta(q, \sigma)! \ \&$$
$$\sigma \text{ is externally enabled}\} \tag{1}$$

We formalize external enablement by a supervisory control function $V : L(\mathbf{G}) \rightarrow Pwr(\Sigma)$, where $Pwr(.)$ stands for power set. For $s \in L(\mathbf{G})$, the evaluation $V(s)$ is defined to be the event subset

$$V(s) := \Sigma_u \cup \{\sigma \in \Sigma_c \mid \sigma \text{ is externally enabled}$$
$$\text{following } s\} \tag{2}$$

In other words, the set (1) is expressible as

$$V(s) \cap \{\sigma \in \Sigma \mid s.\sigma \in L(\mathbf{G})\} \tag{3}$$

namely, the subset of events that, immediately following the generation of s by $\mathbf{G}$, are either enabled by default (executable events in Σ_u) or else by the external controller's decision (a subset of executable events in Σ_c).

It is now easy to visualize how the generating action of $\mathbf{G}$ is restricted by the action of $V(.)$. Initially (having generated the empty string) $\mathbf{G}$ chooses $\sigma_1 \in V(\epsilon)$. Proceeding inductively, after $\mathbf{G}$ has generated $s = \sigma_1.\sigma_2 \ldots \sigma_k \in L(\mathbf{G})$, s is fed back to the controller, which evaluates $V(s)$ according to (2), announcing the result to $\mathbf{G}$, which then chooses σ_{k+1} in (3), and the process repeats. Of course the process would terminate any time the set (3) happened to become empty (although it need not). In any case, we denote the subset of $L(\mathbf{G})$ so determined as $L(V/\mathbf{G})$, called the closed behavior of $V/\mathbf{G}$, where the latter symbol (formally undefined) stands for $\mathbf{G}$ under the supervision of V. It is clear that supervision is a feedback process (Fig. 2), inasmuch as the choice of σ_{k+1} in (3) is not, in general, known in advance, hence must be executed before the succeeding evaluation $V(s.\sigma_{k+1})$ can allow the generating process to continue. With the closed behavior of $V/\mathbf{G}$ now determined, we define the marked behavior

$$L_m(V/\mathbf{G}) := L(V/\mathbf{G}) \cap L_m(\mathbf{G}) \tag{4}$$

namely, those marked strings of $\mathbf{G}$ that survive under supervision by V. Thus supervisory control is nonblocking if $L(V/\mathbf{G}) = \overline{L_m(V/\mathbf{G})}$.
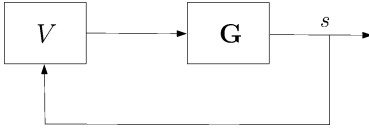
Supervisory Control of Discrete-Event Systems, Fig. 2
Feedback loop $V/\mathbf{G}$

Existence of Controls for DES: Controllability

Of fundamental interest is the question: what sublanguages of $L(\mathbf{G})$ qualify as a language $L(V/\mathbf{G})$ for some choice of supervisory control function V? In other words, what is the scope of controlled behavior(s) for a given $\mathbf{G}$? So far we know that $L(V/\mathbf{G})$ is a sublanguage of $L(\mathbf{G})$, but it is not usually the case that an arbitrary sublanguage would qualify. For instance, the empty string language $\{\epsilon\} \neq L(V/\mathbf{G})$ for any V as in (2) above, in case $\delta(q_o, \sigma)!$ for some σ in Σ_u, for such σ cannot be disabled.

Assume $\mathbf{G}$ is equipped with the technology of controllable events, hence uncontrollable events $\Sigma_u \subseteq \Sigma$. We make the basic definition: the language $K \subseteq \Sigma^*$ is controllable (with respect to $\mathbf{G}$) provided

For all $s \in \overline{K}$ and for all $\sigma \in \Sigma_u$,

whenever $s.\sigma \in L(\mathbf{G})$ then $s.\sigma \in \overline{K}$.   (5)

Informally, a string s can never exit from $\overline{K}$ as the result of the execution by $\mathbf{G}$ of an uncontrollable event: $\overline{K}$ is invariant under the uncontrollable flow. In terms of $\mathbf{G} = \mathbf{MACH}$, above, the languages $\{\epsilon\}, \{wb, wc\}$ are controllable, but $\{w\}, \{w, wcw\}$ are not. For instance, $H := \{w, wcw\}$ has closure $\overline{H} = \{\epsilon, w, wc, wcw\}$, which contains the string $s := w$, but $sb = wb$ can be executed in $\mathbf{MACH}$, b is uncontrollable, and sb has exited from $\overline{H}$. It is logically trivial from (5) that the empty language $\emptyset$ (with no strings whatever) is controllable.

We can now answer the fundamental question posed above.

Given a nonempty sublanguage $K \subseteq L(\mathbf{G})$,

there exists a supervisory control function V

                               (6)

such that $\overline{K} = L(V/\mathbf{G})$, if and only if

K is controllable.

This result exhibits the $L(V/\mathbf{G})$ property in a structured way; furthermore, both the containment $K \subseteq L(\mathbf{G})$ and the controllability property (5) (or its absence) can be effectively (algorithmically) decided in case K itself is the closed or marked behavior of some given DES over Σ.

A key fact easily provable from (5) is that the family of all controllable languages (with respect to a fixed $\mathbf{G}$) is algebraically closed under union, namely,

If K_1 and K_2 are controllable languages,

then so is $K_1 \cup K_2$.   (7)

In fact (7) can be extended to an arbitrary finite or infinite union of controllable languages.

Given $\mathbf{G}$ as above, considered as the plant to be controlled, suppose a new (regular) language E is specified, as the maximal set of strings that we are prepared to tolerate for generation by $\mathbf{G}$; for instance, E could be considered the legal language for $\mathbf{G}$ (irrespective of what $\mathbf{G}$ is potentially capable of generating, namely, $L(\mathbf{G})$). Let us confine attention to the sublanguage of E that contains only marked strings of $\mathbf{G}$, namely, $E \cap L_m(\mathbf{G})$. We now bring in the family $\mathcal{C}(E \cap L_m(\mathbf{G}))$ of all controllable sublanguages of $E \cap L_m(\mathbf{G})$ (including the empty language). From (7) and its infinite extension, there follows the existence of the controllable language

$$K_{\sup} := \cup\{K \mid K \in \mathcal{C}(E \cap L_m(\mathbf{G}))\} \quad (8)$$

We have $K_{\sup} \subseteq E \cap L_m(\mathbf{G})$, and clearly if K' is controllable and $K' \subseteq E \cap L_m(\mathbf{G})$, then $K' \subseteq K_{\sup}$. $K_{\sup}$ is therefore the supremal (largest) controllable sublanguage of $E \cap L_m(\mathbf{G})$. Furthermore, if $K_{\sup}$ is nonempty, then by (6) there exists a supervisory control V such that
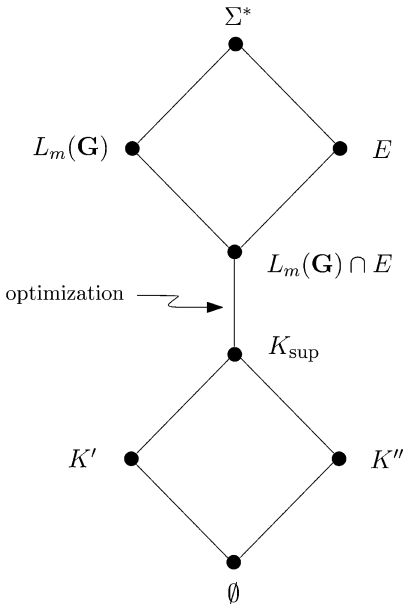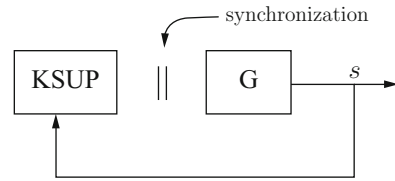
Supervisory Control of Discrete-Event Systems, Fig. 3
Hasse diagram

Supervisory Control of Discrete-Event Systems, Fig. 4
Implementation of $V/\mathbf{G}$

step of the process. In this way the feedback control process is inductively well defined. The computational complexity of this design (cf. (8)) is $O(|\mathbf{E}|^2 \cdot |\mathbf{G}|^2)$ where $\mathbf{E}$ is a DES with $L_m(\mathbf{E}) = E$ and $|\cdot|$ denotes state size. The controller state size is $|\mathbf{KSUP}| \leq |\mathbf{E}| \cdot |\mathbf{G}|$, the product bound being of typical order.

$K_{\text{sup}} = L(V/\mathbf{G})$; in this sense V is optimal (maximally permissive), allowing the generation by $\mathbf{G}$ of the largest possible set of marked strings that the designer considers legal. We have thus established abstractly the existence and uniqueness of an optimal control for given $\mathbf{G}$ and E. This simple conceptual picture is displayed (Fig. 3) as a Hasse diagram, in which nodes represent sublanguages of Σ^* and rising lines (edges) the relation of sublanguage containment.

In a Hasse diagram it could be that K_{sup} collapses to the empty language $\emptyset$. This means that there is no supervisory control for the problem considered, either because the specifications are too severe and the problem is over-constrained or because the control technology is inadequate (more events need to be controllable).

Under the finite-state assumption, K_{sup} is effectively representable by a DES **KSUP**, which may serve as the optimal feedback controller, as displayed in Fig. 4. Here a string s generated by $\mathbf{G}$ drives **KSUP**; at each state of **KSUP**, the events defined in its transition structure are exactly those available to $\mathbf{G}$ for nondeterministic execution (in its corresponding state) at the next

Supervisory Control Design: Small Factory

The following example, Small Factory (SF), is an illustration of supervisor design. As in Fig. 5, SF consists of two machines **MACH1** and **MACH2** each similar to **MACH** above, connected by a buffer **BUF** of capacity 2. In case of breakdown the machines can be repaired by a **SERVICE** facility as shown. Transition structures of the machines and design specifications are also displayed in Fig. 5. Σ_c (Σ_u) are odd (even) numbered events. When self-looped with all irrelevant events to form **BUFSPEC**, the latter specifies that the machines must be controlled in such a way that **BUF** is not overflowed (an attempt by **MACH1** to deposit a workpiece in **BUF** when it is full) or subject to underflow (an attempt by **MACH2** to take a workpiece from **BUF** when it is empty). In addition, **SERVICE** must enforce priority of repair for **MACH2**: when the latter is down, repair of **MACH1** (if in progress) must be interrupted and only resumed after **MACH2** has been repaired; this logic is expressed by **BRSPEC** (appropriately self-looped). To form the plant model $\mathbf{G}$ for the DES to be controlled, we compute the synchronous product of **MACH1** and **MACH2**. The result, say $\mathbf{G} = \mathbf{FACT}$, is a DES of which the components **MACHi** are free
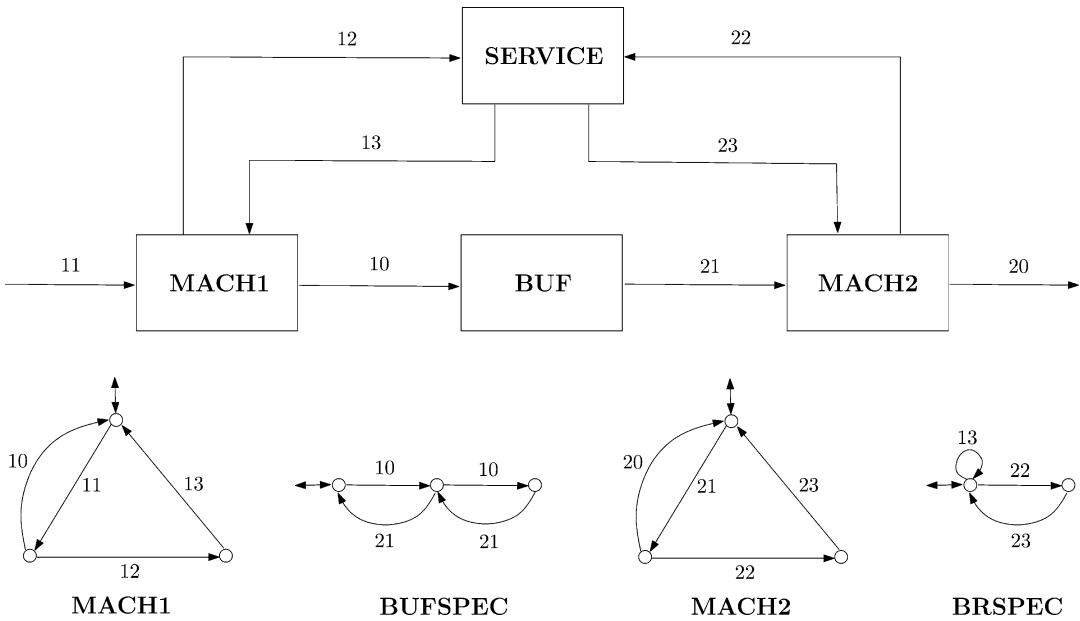
Supervisory Control of Discrete-Event Systems, Fig. 5 Small factory

to execute their events independently except for synchronization on events that are shared (here, none). Similarly we form the synchronous product of **BUFSPEC** and **BRSPEC** to obtain the full specification DES **SPEC**. We now execute the optimization step in the Hasse diagram (Fig. 3); this yields the SF controller **KSUP**(21,47) with 21 states and 47 transitions. Online synchronization of **KSUP** with **FACT** will result in generation of the optimal controlled behavior $K_{\sup}$ by the feedback loop. Since $K_{\sup} \subseteq L_m(\mathbf{G})$ by (8), our marking conventions ensure that **KSUP** is nonblocking.

In general the language $K_{\sup}$ will include in its structure not only the constraints required by control but also the physical constraints enforced by the plant structure itself (here, **FACT**). The latter are thus redundant in the online synchronization of the plant with the controller **KSUP**. A more economical controller is obtained if the plant constraints are projected out of **KSUP** to obtain a reduced controller, say **KSIM**. Mathematically, projection amounts to constructing a control congruence or dynamically (and control) consistent partition on the state set of **KSUP** and taking the cells of this partition, abstractly, as the new

states for **KSIM**. In SF **KSUP** (21,47) is reduced to **KSIM**(5,18), which when synchronized with **FACT** yields exactly **KSUP** but is less than one-quarter the state size. In practice a state size reduction factor of ten or more is not uncommon.

Supervisor Architecture and Computation

As noted earlier, the state size |**KSUP**| of controller **KSUP** is on the order of the product of state sizes of the plant, say |**PLANT**|, and specification, say |**SPEC**|. As these in turn are the synchronous products of individual plant components or partial specifications, |**KSUP**| tends to increase exponentially with the numbers of plant components and specifications, the phenomenon of exponential state space explosion. The result is that centralized or monolithic controllers such as **KSUP** can easily reach astronomical state sizes in realistic industrial models, thereby becoming infeasible in terms of computer storage for practical design. This issue can be addressed in two basic ways: by decentralized and hierarchical architectures, possibly in heterarchical

combination, and by symbolic DES representation and computation, where what is stored are not DES and their controller transition structures in extensional (explicit) form, but instead intensional or algorithmic recipes from which the required state and control variable evaluations are computed online when actually needed.

Supervisory Control Under Partial Observations

Hierarchical control is one example of control under partial observations, a high-level manager (say) observing not full low-level operation but rather an abstraction. Partial observation has been studied mainly for abstractions given by natural projections. For a DES $\mathbf{G}$ over alphabet Σ, let $\Sigma_o \subseteq \Sigma$ be a subalphabet interpreted as the events that can be recorded by some external observer. A mapping $P : \Sigma^* \rightarrow \Sigma_o^*$ is called a natural projection if its action is simply to erase from a string s in Σ^* all the events in s (if any) that do not belong to Σ_o, while preserving the order of events in Σ_o. P extends naturally to a mapping of languages over Σ. One can then implement an induced operator on DES, say Project $(\mathbf{G}) = \mathbf{PG}$, with semantics

$$L_m(\mathbf{PG}) = PL_m(\mathbf{G}), L(\mathbf{PG}) = PL(\mathbf{G}).$$

While in worst cases $|\mathbf{PG}|$ can be exponentially larger than $|\mathbf{G}|$, such blowup seems to be rare, and typically $|\mathbf{PG}| \leq |\mathbf{G}|$, namely, P results in simplification of the model $\mathbf{G}$. By use of P it is possible to carry over to DES the control-theoretic concept of observability. Two strings $s, s' \in \Sigma^*$ are look-alikes with respect to P if $Ps = Ps'$, namely, are indistinguishable to an observer (or channel) modeled by P. Thus, given $\mathbf{G}$ and P as above, a sublanguage $K \subseteq L(\mathbf{G})$ is observable if, roughly, look-alike strings in $\overline{K}$ have the same one-step extensions in $\overline{K}$ that are compatible with membership in $L(\mathbf{G})$ and also satisfy a consistency condition with respect to membership in $L_m(\mathbf{G})$. For control under observations through P, one defines a supervisory

control function $V : L(\mathbf{G}) \rightarrow Pwr(\Sigma)$ to be feasible if it assumes the same value on look-alike strings, in other words respects the observation constraint enforced by P. It then turns out that a language $K \subseteq L_m(\mathbf{G})$ can be synthesized in a feedback loop including $\mathbf{G}$ and the feedback channel P if and only if K is both controllable and observable.

Although this result is conceptually satisfying, it is computationally inconvenient because, by contrast with controllability, the property of sublanguage observability is not in general closed under union. A substitute for observability is sublanguage normality, a stronger property than observability but one that is indeed closed under union. Since the family of controllable and normal sublanguages of a given specification language is nonempty (the empty language belongs) and is closed under union, a (unique) supremal (or optimal) element exists and can be computed; it therefore solves the problem of supervisory control under partial observations, albeit under the normality restriction. The latter has the feature that the resulting supervisor can only disable a controllable event if the latter is observable, i.e., belongs to Σ_o. In some applications this restriction might preclude the existence of a solution altogether; in others it could be harmless, or even desirable as a safety property, in that if the intended disablement of a controllable event happened to fail, and the event occurred after all, the fault would necessarily be observable and thus optimistically remediable in good time.

An intermediate property is known that is weaker than normality but stronger than observability, called relative observability. The family of relatively observable sublanguages of a given specification language is closed under union and thus does possess a supremal element, which in the regular case can be effectively computed. When combined with controllability, relative observability yields a solution to the problem of supervisory control under partial observations which places no limitation on the disablement of unobservable controllable events. Examples show that a nontrivial solution of this type may exist in cases where the normality solution is empty.

Summary and Future Directions

Supervisory control of discrete-event systems, while relatively new, has reached a first level of maturity in that it is soundly based in a standard framework of (especially) finite-state machines and regular languages. It has effectively incorporated its own versions of control-theoretic concepts like stability (in the sense of nonblocking), controllability, observability, and optimality (in the sense of maximal permissiveness). Modular architectures and, on the computational side, symbolic approaches enable design of both monolithic and heterarchical/distributed controllers for DES models of industrial size. Major challenges remain, especially to develop criteria by which competing architectures can be meaningfully compared and to organize control functionality in ways that are not only tractable but also transparent to the human user and designer.

Cross-References

▶ Applications of Discrete-Event Systems
▶ Models for Discrete Event Systems: An Overview

Bibliography

Cassandras CG, Lafortune S (2008) Introduction to discrete event systems. Springer, New York
Cieslak R, Desclaux C, Fawaz A, Varaiya P (1988) Supervisory control of discrete-event processes with partial observations. IEEE Trans Autom Control 33:249–260
Lin F, Wonham WM (1988) On observability of discrete-event systems. Inf Sci 44:173–198
Ma C, Wonham WM (2005) Nonblocking supervisory control of state tree structures. Lecture notes in control and information sciences, vol 317. Springer, Berlin
Ramadge PJ, Wonham WM (1987) Supervisory control of a class of discrete event processes. SIAM J Control Optim 25:206–230
Seatzu C et al (ed) (2013) Control of discrete-event systems. Springer, London/New York
Wonham WM (1997–2013) Supervisory control of discrete-event systems. Department of Electrical and Computer Engineering, University of Toronto. Available at http://www.control.utoronto.ca/~wonham

Switching Adaptive Control

Minyue Fu
School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW, Australia

Abstract

Switching adaptive control is one of the advanced approaches to adaptive control. By employing an array of simple candidate controllers, a properly designed monitoring function and switching law, this approach is capable to search in real time for a correct candidate controller to achieve the given control objective such as stabilization and set-point regulation. This approach can deal with large parameter uncertainties and offers good robustness against unmodelled dynamics. This article offers a brief introduction to switching adaptive control, including some historical background, basic concepts, key design components, and technical issues.

Keywords

Adaptive control; Hybrid systems; Multiple models; Supervisory control; Switching logic; Uncertain systems

Introduction

Switching adaptive control, also known as *switched adaptive control* or *multiple model adaptive control,* refers to an *adaptive control* technique which deploys a set of controllers and a switching law to achieve a given control objective. The concept of switching adaptive control is generalized from the traditional *gain scheduling* technique (Leith and Leithead 2000). As in the standard adaptive control setting, the model for the controlled plant is assumed to contain uncertain parameters, and the control objective is to stablize the system and, in many cases, to deliver certain performance using

real-time information in the measured output. What differentiates switching adaptive control from gain scheduling is that the uncertain parameters are not directly measured and the switching is determined by the system response. This seemingly minor difference is very important because parameter estimation may not be possible due to the lack of persistent excitation; moreover, the sensitivity of the measured output is often suppressed by the feedback control which makes closed-loop identification of the uncertain parameters difficult. Compared with classical adaptive control, switching adaptive control has better inherent robustness against parameter uncertainties and unmodelled dynamics.

By early 1980s, the classical adaptive control theory for linear systems had been well established under a set of so-called classical assumptions, which include:

- Known order of the plant (or known maximum order of the plant)
- Known relative degree of the plant
- Minimum phase dynamics
- Known sign of the high-frequency gain (which is the gain of the plant when the input is high-frequency sinusodial signal)

At the same time, it was recognized that the classical adaptive control approach has inherent robustness problems against even miniature unmodelled dynamics (Rohrs et al. 1985). While this generated a wave of research aiming at robustification of the classical adaptive control theory (see, e.g., Ioannou and Sun 1996), a new line of research took place aiming at relaxing the classical assumptions. Nussbaum (1983) paved the way by showing that knowledge of the sign of the high-frequency gain can be avoided for a first order linear system. Morse (1985) developed a "universal controller" which can adaptively stablize any strictly proper, minimum-phase system with relative degree not exceeding two. Martensson (1985) gave a very surprising result by showing that asymptotic stabilization can be achieved adaptively by simply assuming that there exists a finite order stabilizer. But Martensson's controller is impractical due to the need for exhaustive online search of the stabilizer and subsequent excessively high overshoots. Switching adaptive

control was then introduced in Fu and Barmish (1986), aiming at achieving adaptive stabilization with minimal assumptions and a guarantee of exponential convergence rate for the state. In contrast to the work of Martensson, a compactness requirement is made on the set of possible plants and an upper bound on the order of the plant is assumed. These assumptions allow a set of possible plants to be partitioned into a finite number of subsets, with each stabilizable by a single controller. A monitoring function and a switching law are then designed to sequentially eliminate incorrect candidate controllers until an appropriate controller is found. Due to the fact that the number of candidate controllers may be large, many follow-up works on switching adaptive control focused on speeding up the switching process by eliminating incorrect candidate controllers without trying them (Zhivoglyadov et al. 2000, 2001). These results can also deal with slowly time-varying parameters and infrequent parameter jumps.

Another major breakthrough came from the works of Morse (1996, 1997) under the term of *supervisory control*. His work considers set-point regulation for uncertain linear systems. A different compactness requirement is used to allow unmodelled dynamics in the system. More specifically, the given uncertain linear system is assumed to belong to a union of sub-families of systems, with each sub-family having a linear controller capable to achieve set-point regulation. Suitably defined output-squared estimation errors are used as monitoring functions and a candidate controller is selected whose corresponding performance signal is the smallest. The major advantages of this switching law are that the "correct" controller can usually be quickly identified without cycling through all possible candidate controllers, leading to a good closed-loop performance.

More recent research on switching adaptive control focuses on more systematic and alternative approaches to the design of candidate controllers and switching laws; see, e.g., Anderson et al. (2000), Hespanha et al. (2001), and Morse (2004). Generalizations to nonlinear systems are also found Battistelli et al. (2012).

Design of Switching Adaptive Control

A switching adaptive controller consists of the following key ingredients:

- Design of control covering
- Design of monitoring function
- Selection of dwell time

For illustrative purposes, we consider an adaptive stabilization problem where the system has the following model:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t)$$

with state $x(t) \in R^n$ for some $1 \leq n \leq n_{\max}$ and the measured output $y(t) \in R^r$. The given set of uncertain plants Σ consits of triplets (A, B, C) and we use the notation $\Sigma^{(n)}$ to denote the subset of Σ consisting of those plants having order n. It is assumed that every possible plant $(A, B, C) \in \Sigma$ is a minimal realization (i.e., both controllable and observable) and that every $\Sigma^{(n)}$ is a compact set (i.e., it is closed and bounded). The control objective is to design an adaptive controller to drive the state to zero asymptotically, i.e., $x(t) \rightarrow 0$ as $t \rightarrow \infty$. It is clear that each possible plant in Σ admits a linear dynamic stabilizer. An alternative description of the uncertain plant is introduced in Morse (1996, 1997) where its transfer function is a member of a continuously parameterized set of admissible transfer functions of the form

$$\Sigma \subset \bigcup_{p \in \mathcal{P}} \{v_p + \delta : \|\delta\| \leq \varepsilon_p\}$$

In the above, $\mathcal{P}$ is a compact set in a finite dimensional space, v_p is a nominal transfer function with its coefficients depending continuously on p, δ is the transfer function of some unmodelled dynamics, $\|\delta\|$ represents a shifted H_∞ norm (obtained by first shifting the poles of δ slightly to the right and then computing its H_∞ norm), and ε_p is sufficiently small so that each set of plants $\{v_p + \delta : |\delta| \leq \varepsilon\}$ is stabilizable by a single controller for all $p \in \mathcal{P}$.

Control covering: The purpose is to decompose the given set of plants into a union of subsets such that each subset P_i admits a single controller K_i (called candidate controller) to achieve the given control objective. This is typically done using two properties: inherent robustness of linear controllers and the existence of a finite cover for any compact set. More specifically, if a candidate controller renders a desired control objective for a given plant, then the same objective is maintained when the plant is perturbed slightly. For example, Fu and Barmish (1986) uses the fact that if a given plant is stabilized by a controller then the same controller stabilizes all the plants with sufficiently small parameter perturbations. Similarly, Morse (1996, 1997) uses the fact that the same controller achieves set-point regulation for a small neighborhood of plants. Combining this property with the finite covering property yields

$$\Sigma = \bigcup_{i=1}^{N} \Sigma_i$$

such that each subset Σ_i admits a single controller K_i.

Monitoring Function: The generation of the adaptive switching controller is accomplished using a *switching law* or *switching logic* whose task is to determine, at each time instant, which candidate controller is to be applied. The core of the switching law is a monitoring function. Its very basic role is to be able to detect whether the applied candidate controller is consistent with the corresponding plant subset so that wrong candidate controllers can be eliminated one by one until an appropriate controller is found. A major difficulty for switching adaptive control design is that persistent excitation is not assumed. Consequently, it is not always possible to detect the correct plant subset using the measured output. The key idea is to check which plant subsets are consistent with the generated output.

One simple monitoring function uses a finite-time L_2 norm of the measured output:

$$V(t, \tau) = \int_{t-\tau}^{t} \|y(s)\|^2 \, ds$$

where τ is the so-called *dwell time*. It turns out that for some properly chosen dwell time, a correctly applied candidate controller is able to guarantee some decay property for the monitoring function, i.e., $V(t, \tau) \leq e^{-\lambda \tau} V(t - \tau, \tau)$ for some $\lambda > 0$. This property is sufficient to allow a wrong candidate controller to be eliminated. However, much smarter monitoring functions can be designed so that infeasible candidate controllers (those not corresponding to the true plant) can be eliminated without even being applied. This can be done using the *falsification* approach in parameter estimation where the basic idea is to eliminate all plant subsets Σ_i inconsistent with the measured output signal. For example, consider the following discrete-time model:

$$y(t) = -a_1 y(t-1) - a_2 y(t-2)$$
$$+ b_1 u(t-1) + b_2 u(t-2) + w(t)$$

where a_i and b_i are uncertain parameters and w(t) is a bounded disturbance, i.e., $|w(t)| \leq \delta$ for some δ. For this example, we may eliminate all the uncertain parameter subsets which violate the following constraint (Zhivoglyadov et al. 2000):

$$|y(t) + a_1 y(t-1) + a_2 y(t-2)$$
$$- b_1 u(t-1) - b_2 u(t-2)| \leq \delta$$

More generally, one can use the so-called multi-estimator (Morse 1996, 1997) which involves an array of estimators, one for each plant subset Σi using its nominal model. The output estimation error $e, (l)$ for each such estimator is then used to construct a monitoring function, e.g.,

$$V_i(t, \tau) = \int_{t-\tau}^{t} e^{-2\lambda(t-s)} \|e_i(s)\|^2 \, ds$$

where τ is the dwell time as before and $\lambda > 0$ is an exponential weighting parameter used to guarantee the decay rate of the monitoring function as before. Instead of using the monitoring functions to eliminate infeasible candidate controllers, the candidate controller corresponding to the least estimation error, as measured by the least monitoring function, is selected. The main advantage of the multi-estimator based monitoring functions is that falsification of candidate controllers is done implicitly and a "correct" controller can be quickly reached, leading to good performance.

Dwell Time: The dwell time τ as defined above is a critical component in switching adaptive control. Serving in the monitoring function, this is the minimum nonzero amount of time for a candidate controller to be applied before switching. That is, this provides a sufficient time lag to build the monitoring function so that its exponential decay property is detected when a correct candidate controller is applied. This will allow detection of infeasible plant subsets and selection of a "correct" controller. The use of a dwell time also avoids arbitrarily fast switching, thus gaurantee-ing the solvability of the system dynamics.

The dwell time can be selected a priori by using the fact that if a matrix A is stable, then there exist some positive values λ and τ such that $\|e^{At}\| \leq e^{-\lambda \tau}$ for all $i > \tau$. This leads to the desired exponential decaying property

$$V(t, \tau) \leq e^{-\lambda \tau V}(t - \tau, \tau)$$

for the aforementioned monitoring function for adaptive stabilization.

Alternatively, the dwell time can be chosen implicitly. Hespanha et al. (2001) suggest a *hysteresis switching logic* method. This method employs a hysteresis parameter $h > 0$. Suppose the candidate controller K_j is applied at time t_i, then K_j is kept until the next switching time t_{i+1} which is the minimum $t \leq t_i$, such that

$$(1 + h) \min_{1 \leq k \leq N} V_k(t, t - t_i) \leq V_j(t, t - t_i)$$

Because h > 0, the time difference $t_{l+1} - t_i > 0$ is lower bounded, which implies the existence of a dwell time.

Summary and Future Directions

Switching adaptive control is a conceptually simple control technique capable to deal with large

S

parameter uncertainties. The use of simple candidate controllers (typically linear) imply good closed-loop behavior and good robustness against unmodelled dynamics. Although the discussion above assumes that the number of plant subsets is finite, this assumption is not essential; see Anderson et al. (2000).

Switching adaptive control renders the closed-loop system a switched system or hybrid system, for which a wide range of tools are available to aid the analysis of such a system; see, e.g., Liberzon (2003). However, unique features of such a system arise from the fact that the switching mechanism is chosen by the designer, rather than being a part of the given plant. How to best design the switching mechanism is an interesting issue.

Future works for switching adaptive control include:

1. How to simplify the design of candidate controllers. Finite covering based design often yields a large number of plant subsets, hence a large number of candidate controllers. Since most of the candidate controllers do not need to apply (which is the case when falsification based switching logic is used, for example), smarter ways are needed for the design of candidate controllers.

2. Wider applications. Most of the research so far focuses on stabilization and set-point regulation (which is essentially a stabilization problem). How to incorporate general performance criteria is an essential and yet challenging issue.

3. Better design of monitoring functions and the corresponding switching logic. Most existing monitoring functions use a finite-time L_2 norm of the output (or regulation error), with the key feature that some exponential decay property is guaranteed when the candidate controller is "correct." Note that the key purpose of the monitoring function and the corresponding switching logic is to allow fast falsification of infeasible candidate controllers. Thus, a much wider range of monitoring functions can possibly be used. In particular, how to incorporate set membership identification techniques (Milanese and Taragna 2005) may be of particular interest.

Cross-References

▶ Adaptive Control, Overview
▶ Hybrid Dynamical Systems, Feedback Control of
▶ Robust Model-Predictive Control
▶ Stability and Performance of Complex Systems Affected by Parametric Uncertainty

Bibliography

Anderson BDO, Brinsmead T, Bruyne FD, Hespanha JP, Liberzon D, Morse AS (2000) Multiple model adaptive conrol. Part 1: finite controller coverings. Int J Robust Nonlinear Control 10(11–12):909–929

Battistelli G, Hespanha JP, Tesi P (2012) Supervisory control of switched nonlinear systems. Int J Adapt Control Signal Process 26(8):723–738. Special issue on Recent Trends on the Use of Switching and Mixing in Adaptive Control

Fu M, Barmish BR (1986) Adaptive stabilization of linear systems via switching control. IEEE Trans Autom Control 31(12):1097–1103

Hespanha JP, Liberzon D, Morse AS, Anderson BDO, Brinsmead T, Bruyne FD (2001) Multiple model adaptive control. Part 2: switching. Int J Robust Nonlinear Control 11:479–496

Ioannou P, Sun J (1996) Robust adaptive control. Prentice Hall, Upper Saddle River

Leith DJ, Leithead WE (2000) Survey of gain-scheduling analysis and design. Int J Control 73(11):1001–1025

Liberzon D (2003) Switching in systems and control. Birkhäuser, Boston

Martensson B (1985) The order of any stabilizing regulator is sufficient information for adaptive stabilization. Syst Control Lett 6:87–91

Milanese M, Taragna M (2005) H-infinity set membership identification: a survey. Automatica 41:2019–2032

Morse AS (1985) A three-dimensional universal controller for the adaptive stabiliztion of any strictly proper minimum-phase system with relative degree not exceeding two. IEEE Trans Autom Control 30(12):1188–1191

Morse AS (1996) Supervisory control of famillies of linear set-point controllers part I: exact matching. IEEE Trans Autom Control 41(10):1413–1431

Morse AS (1997) Supervisory control of families of linear set-point controllers part II: robustness. IEEE Trans Autom Control 42(11):1500–1515

Morse AS (2004) Lecture notes on logically switched dynamical systems. In: Nistri P, Stefani G (eds) Nonlinear and optimal control theory. Springer, Berlin, pp 61–162

Nussbaum RD (1983) Some remarks on a conjecture in parameter adaptive control. Syst Control Lett 3:243–246

Rohrs CE, Valavani L, Athans M, Stein G (1985) Robust-
    ness of continuous-time adaptive control algorithms
    in the presence of un-modeled dynamics. IEEE Trans
    Autom Control 30(9):881–889

Zhivoglyadov PV, Middleton RH, Fu M (2000) Localiza-
    tion based switching adaptive control for time-varying
    discrete-time systems. IEEE Trans Autom Control
    45(4):752–755

Zhivoglyadov PV, Middleton RH, Fu M (2001) Further re-
    sults on localization based switching adaptive control.
    Automatica 37:257–263

Synthesis Theory in Optimal Control

Ugo Boscain[1,2] and Benedetto Piccoli[3]
[1]CNRS CMAP, École Polytechnique, Palaiseau,
France
[2]Team GECO INRIA Saclay, Palaiseau, France
[3]Mathematical Sciences and Center for
Computational and Integrative Biology, Rutgers
University, Camden, NJ, USA

Abstract

In this entry we review the theory of optimal
synthesis. We describe the steps necessary to
solve an optimal control problem and the suf-
ficient conditions for optimality given by the
theory. We describe some relevant examples that
have important applications in mechanics, in the
theory of hypo-elliptic operators and for the study
of models of geometry of vision. Finally, we
discuss the problem of optimal stabilization and
the difficulties encountered if one tries to give the
solution to the problem in feedback form.

Keywords

Affine control systems; Extremals; Pontryagin
Maximum Principle; Sub-Riemannian geometry;
Time-optimal synthesis

Optimal Control

An optimal control problem with fixed initial and
terminal conditions can be seen as a problem
of calculus of variations under nonholonomic
constraints:

$$\dot{q}(t) = f(q(t), u(t)), \tag{1}$$

$$\int_0^T L(q(t), u(t))\, dt \rightarrow \min \quad (T \text{ fixed or free}), \tag{2}$$

$$q(0) = q_0, \quad q(T) = q_1. \tag{3}$$

Here we make the following set of assumptions:

(H) q belongs to a finite-dimensional smooth
manifold M of dimension n. As a function of
time $q(.)$ is assumed to be Lipschitz continuous.
The control $u(.)$ is a L^∞ function taking values
in a set $U \subset \mathbb{R}^m$. For simplicity, we assume that
the functions f and L, defined on $M \times \mathbb{R}^m$, are
smooth.

The dynamics $\dot{q}(t) = f(q(t), u(t))$ play the
role of the nonholonomic constraint (nonholo-
nomic means that it is a constraint on the velocity
but not necessarily on the position).

Solving an optimal control problem in general
is a very difficult task. Usually, to attack such a
problem, the steps are the following:

- STEP 0: EXISTENCE. First, one has to guar-
  antee the existence of a solution to (1)–(3).
  The most important sufficient condition for
  the existence of minimizers is the famous
  Filippov theorem (see for instance Agrachev
  and Sachkov (2004) for a proof) saying the
  following: introduce a new variable (the so-
  called augmented state) $\hat{q} := (q^0, q) \in
  \mathbb{R} \times M$ satisfying the following dynamics:

$$\dot{\hat{q}}(t) = \begin{pmatrix} \dot{q}^0(t) \\ \dot{q}(t) \end{pmatrix} = \begin{pmatrix} L(q(t), u(t)) \\ f(q(t), u(t)) \end{pmatrix}$$

$$=: \hat{f}(\hat{q}(t), u(t)) \tag{4}$$

then if (i) U is compact; (ii) the set of ve-
locities $F(\hat{q}) := \{\hat{f}(\hat{q}, u) \mid u \in U\}$ is
convex for every $\hat{q}$; (iii) for every $T > 0$ and
$\hat{q}_0 \in \mathbb{R} \times M$, there exists a compact set $K \subset
\mathbb{R} \times M$ such that all solutions of (4) starting
from $\hat{q}_0$ stay in K for $t \in [0, T]$; then there
exist Lipschitz minimizers. Other theorems
that can be applied in more general functional
classes or under less restrictive hypotheses can

S

be found in the literature. See for instance Bressan and Piccoli (2007), Cesari (1983), and Vinter (2010).

- STEP 1: FIRST ORDER NECESSARY CONDITIONS. In optimal control, the first order necessary conditions for optimality are given by the celebrated Pontryagin Maximum Principle (Pontryagin et al. 1961) (see also Agrachev and Sachkov (2004) for a more recent viewpoint). The Pontryagin Maximum Principle (PMP for short) extends the (Hamiltonian version of the) Euler-Lagrange equations of calculus of variations to problems with nonholonomic constraints. For a discussion about the relation between variational problems under nonholonomic constrains and variational principles in nonholonomic mechanics, see Bloch (2003).

The PMP restricts the set of candidate optimal trajectories starting from q_0 to a family of trajectories, called *extremals*, parameterized by a covector $p(0) \in T_{q_0}^* M$. In addition, there are two kinds of special extremals: (i) the *singular extremals* for which the maximization condition given by the PMP does not permit directly obtaining the control and (ii) the *abnormal extremals* which are candidate optimal trajectories for any cost function. For certain classes of problems, abnormal extremals and singular trajectories coincide.

The set of all trajectories satisfying the PMP (in general having intersections and not being all optimal forever) is called an *extremal synthesis*. The requirement that the trajectories starting from q_0 reach the final point q_1 (at time T, fixed or free) is usually not very useful at this step. This requirement is rather made at STEP 4.

- STEP 2. HIGHER ORDER CONDITIONS. Higher order conditions are used to restrict further the set of candidate optimal trajectories. The most important conditions are those used to eliminate singular extremals (which usually are very hard to treat) as the Goh condition and the generalized Legendre-Clebsch conditions (see for instance Agrachev and Sachkov 2004). Other theories that provide higher order conditions (which apply

also to extremals that are not singular) are for instance: higher order maximum principles (Bressan 1985; Krener 1977), generalized Morse-Maslov index theories (Agrachev and Sachkov 2004), and envelope theory (Sussmann 1986, 1989, see also Boscain and Piccoli 2004, Cap. 1.3.2).

- STEP 3. SELECTION OF THE OPTIMAL TRAJECTORIES. This step is the most difficult one. Indeed, one should check that each extremal of the extremal synthesis does not intersect another extremal having a smaller cost at the intersection point. This comparison should be done not only among extremals which are close, one to the other, but among all of them. The problem is indeed global.

One of the techniques to address this problem in a very elegant way takes the name of *optimal synthesis theory*, and was developed almost together with the birth of the Pontryagin Maximum Principle. This theory dates back to the paper of Boltyanskii (1966) and was further developed by Brunovsky (1980, 1978), Sussmann (1980, 1979), and Piccoli and Sussmann (2000).

Roughly speaking, the theory of optimal synthesis permits to conclude that if one has an extremal synthesis having certain regularity properties, then this extremal synthesis is indeed an optimal synthesis.

An optimal synthesis is a collection of optimal trajectories starting from q_0 and reaching the various points of the space:

$$\mathcal{S}_{q_0} = \{\gamma_q(.) : [0, T_q] \to M \,|\, q \in M, \gamma_q \text{ is}$$
a trajectory of (1) minimizing the cost $\int_0^{T_q} L(q(t), u(t)) \, dt$ with $\gamma(0) = q_0, \ \gamma(T) = q\}$

An optimal synthesis should also verify the following condition: if γ_q defined on $[0, T]$ and γ_q' defined on $[0, T']$ (with $T' \in]0, T[$) belong to $\mathcal{S}_{q_0}$ and we have $q' = \gamma_q(T')$ then $\gamma_{q'} = \gamma_q|_{[0,T']}$. More details are given in the next section.

- STEP 4. SELECTION OF THE TRAJECTORY REACHING THE FINAL POINT. Once an optimal synthesis is computed,

one selects the optimal trajectory reaching the desired final point solving the equation $\gamma(T) = q_1$, in the set of all trajectories belonging to the optimal synthesis.

Remark 1 Notice that one could require that the final point is reached at STEP 1. This would considerably reduce the set of candidate optimal trajectories already at STEP 1, but would not permit to apply the powerful (global) theorems of STEP 3. As a consequence, one would be obliged to compare by hands all extremals going from q_0 to q_1.

Sufficient Conditions for Optimality: The Theory of Optimal Synthesis

There exists a general principle for which every synthesis formed by extremals is optimal under very mild regularity conditions. We will illustrate a classical case of a feedback smooth on a stratification, due to Boltianskii and Brunovsky, see Boltyanskii (1966) and Brunovsky (1980, 1978). More general results can be found in Piccoli and Sussmann (2000). This principle is very strong and is valid only because the synthesis is a global object, while given a single trajectory satisfying PMP, there is no regularity condition which ensures optimality.

For simplicity, from now on, we assume that $M = \mathbb{R}^n$ is an Euclidean space and $q_0 = 0$ and indicate by $\mathcal{S}$ a candidate optimal synthesis from 0, the general case follows easily. A set $P \subset M$ is said a *curvilinear open polytope* of dimension p, if there exists a polytope (i.e., bounded closed region intersection of a finite number of half-spaces) $P' \subset \mathbb{R}^p$ and a smooth map $\phi : \mathbb{R}^p \to \mathbb{R}^n$, injective with jacobian having maximal rank at every point, such that $\phi(P' \setminus \partial P') = P$.

Let Ω be an open subset of M (for the induced topology) containing the origin in its interior. We say that $\mathcal{S}$ is a *Boltyanskii–Brunovsky regular synthesis*, briefly BB synthesis, if the following holds.

There exists a 6–tuple $\Xi = (\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2, \prod, \Sigma, u)$ such that

(BB1) $\mathcal{P}$ is a collection of curvilinear open polyhedra and Ω is disjoint union of elements of $\mathcal{P}$. If $P_j \neq P_k \in \mathcal{P}$ and $P_k \cap \overline{P_j} \neq \emptyset$ then $P_k \subset \partial P_j$ and $\dim(P_k) < \dim(P_j)$. $\{0\} \in \mathcal{P}$ and the elements of $\mathcal{P}$ are called "cells".

(BB2) $\mathcal{P} \setminus \{\{0\}\}$ is the disjoint union of $\mathcal{P}_1$ (the set of "type I cells") and $\mathcal{P}_2$ (the set of "type II cells"),

(BB3) the feedback $u : \{q : \exists P_1 \in \mathcal{P}_1, q \in P_1\} \to U$ and $\prod : \mathcal{P}_1 \to \mathcal{P}$ are maps, $\Sigma : \mathcal{P}_2 \to \mathcal{P}_1$ is a multifunction, with non empty values, such that the following properties are satisfied:

(i) The function u is of class $\mathcal{C}^1$ on each cell.

(ii) If $P_1 \in \mathcal{P}_1$, then $f(q, u(q)) \in T_q P_1$ (the tangent space to P_1 at q) for every $q \in P_1$. In addition, for each $q \in P_1$, if we let ξ_q be the maximally defined solution to the initial value problem

$$\dot{\xi} = f(\xi, u(\xi)), \quad \xi(0) = x, \quad \xi \in P_1, \tag{5}$$

and define $t_q = \sup Dom(\xi_q)$, then the limit $\xi_q(t_q-) := \lim_{t \uparrow t_q} \xi_q(t)$ exists and belongs to $\prod(P_1)$.

(iii) If $P_2 \in \mathcal{P}_2$, then for each $q \in P_2$ and $P \in \Sigma(P_2)$ there exists a unique curve $\xi_q^P : [0, t_q^P[\to \Omega$ such that the restriction of ξ_q^P to $]0, t_q^P]$ is a maximally defined integral curve of the vector field $f(\cdot, u(\cdot))$ on P, and $\xi_q^P(0) = q$.

(iv) On every cell $P_1 \in \mathcal{P}_1$, $q \to t_q$ is a continuously differentiable function, and $(t, q) \to \xi_q(t)$, $(t, q) \to u_q(t) := u(\xi_q(t))$ are continuously differentiable maps on the set

$$E(P) := \{(t, q) : q \in P_1, t \in [0, t_q]\}.$$

If $P_2 \in \mathcal{P}_2$ the same holds for every t_q^P, ξ_q^P, u_q^P, with $P \in \Sigma(P_2)$.

(v) For every $q \in \Omega \setminus \{0\}$, the trajectory $\gamma_q : [0, T_q] \to M, \gamma_q \in \mathcal{S}$, is obtained by piecing together the trajectories on every single cell. Moreover, γ_q changes cell a finite number of times.

S

Theorem 1 (Sufficiency theorem for BB synthesis) *Let $\mathcal{S}$ be a BB synthesis on M formed by extremal trajectories, then $\mathcal{S}$ is optimal.*

Remark 2 Theorem 1 can be proved also for synthesis on an open subset Ω of M, under suitable conditions, see Piccoli and Sussmann (2000).

Some Relevant Examples

Even if the sufficient conditions for optimality given by the theory of optimal synthesis are very powerful, in general computing explicitly an optimal synthesis is very hard and the complexity grows quickly with the dimension of the space. The main difficulties are:

- The integration of the Hamiltonian equations given by the PMP (which in general is not integrable, unless there are many symmetries);
- The characterisation of singular and abnormal extremals;
- The verification of the hypotheses of the sufficient conditions for optimality given by synthesis theory.

For these reasons, the computation of optimal synthesis is already challenging in dimension 2, and few examples have been solved in dimension 3. In higher dimensions, only very symmetric problems have been completely solved. In the following, we list some of the most relevant optimal synthesis that have been computed up to now.

Time-Optimal Synthesis for Affine Control Systems on 2-D Manifolds

Let M be a 2-D manifold and consider the problem of finding the time-optimal synthesis starting from a point q_0 for a system of the type

$$\dot{q} = F(q) + uG(q), \quad |u| \leq 1, \quad F(q_0) = 0 \tag{6}$$

Here we assume that F and G are Lie-bracket generating. The condition $F(q_0) = 0$ guarantees local controllability around q_0, for a generic pair (F, G). A complete theory for this kind of systems, was developed in Bressan and Piccoli (1998), Piccoli (1996), and Boscain and Piccoli (2004), under generic conditions on the vector fields F and G. More precisely, in Boscain and Piccoli (2004) it was provided: (i) an algorithm building explicitly the time-optimal synthesis; (ii) a classification of synthesis in terms of graphs; (iii) a classification of synthesis singularities; (iv) an analysis of the properties of the minimum time function.

Here we just recall that optimal trajectories are a finite concatenation of bang (trajectories corresponding to constant control $+1$ or -1) and singular arcs (for which the control may correspond to something different from $+1$ or -1).

Under generic conditions, the optimal synthesis provides a stratification of M. In the regions of dimension 2, the control is either $+1$ or -1. The regions of dimension 1 called *Frame Curves* can be: (i) arcs of optimal trajectories (that may be bang or singular); (ii) switching curves (i.e., curves made of points in which the control switches from $+1$ or -1, or viceversa); (iii) overlap curves (i.e., curves made of points where the extremals lose their optimality). The region of dimension 0 called *Frame Points* are points where frame curves intersect. Generically, they can be of 23 types. See Boscain and Piccoli (2004, p. 60).

Some Relevant Time-Optimal Synthesis for 3D Problems

As we saw in the previous section, for minimum time problems in dimension 2, many results can be obtained, and in most cases a time-optimal synthesis can be constructed. The situation is different for time-optimal problems in dimension 3. Indeed, beside trivial cases, the time-optimal synthesis was computed in full details for few examples only. One is the Reed and Shepp's car,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = u_1 \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad |u_1|, |u_2| \leq 1. \tag{7}$$

The time-optimal synthesis for this problem was computed in Soueres and Laumond (1996). The extreme complexity of the optimal synthesis obtained for this simple example had the effect that no other time-optimal synthesis in dimension 3 or larger, with one or two bounded controls, were computed up to the last 2-years.

Very recently, the interest in time-optimal synthesis for systems of the type

$$\dot{q} = \sum_{i=1}^{m} u_1 F_i(q), \quad |u_i| \leq 1, \quad (i = 1, \ldots, m) \tag{8}$$

where q belongs to a n-dimensional manifold and $2 \leq m \leq n$, has attracted new attention.

This is indeed a problem of nonstrictly convex sub-Finsler geometry that appears in the study of asymptotic cones of nilpotent groups in geometric group theory (Gromov 1981; Breuillard and Le Donne 2012).

Sub-Riemannian Geometry

A very important class of optimal control problems is the one called sub-Riemannian. Let M be a n-dimensional manifold ($n \geq 2$) and consider the problem of finding the time-optimal synthesis starting from a point q_0 for the problem

$$\dot{q} = \sum_{i=1}^{m} u_i F_i(q), \quad \int_0^1 \sqrt{\sum_{i=1}^{m} u_i^2} \, dt \to \min,$$
$$(2 \leq m \leq n) \tag{9}$$

Here we assume that the family of vector fields $\{F_i\}_{i=1\ldots m}$ is Lie-bracket generating. This kind of optimal-control problems includes Riemannian geometry and many of its generalizations that usually take the name of sub-Riemannian geometry (see Bellaiche (1996), Montgomery (2002) and the pioneering work by Brockett (1982)). The complete time optimal synthesis was computed in a few relevant cases:

- The Heisenberg group (Gaveau 1977; Gershkovich and Vershik 1988).
- The local 3-dimensional contact case, under generic conditions (Agrachev 1996; El-Alaoui et al. 1996).

- Some relevant left-invariant problem on simple Lie groups, i.e., $SO(3)$, $SU(2)$, $Sl(2)$, see Boscain and Rossi (2008).
- The left-invariant problem on the group of rototranslation $SE(2)$ that has important applications in models of geometry of vision (Boscain et al. 2012; Sachkov 2011; Petitot 2008).
- In dimension bigger than 3, only the quasi-Heisenberg case (Charlot 2002) and certain multidimensional generalizations of the Heisenberg case has been computed (Beals et al. 1996).
- In dimension 2, problems of type (8) are called problems of *almost-Riemannian geometry*. The basic example (the so-called Grushin case) was studied in Bellaiche (1996) and the study of the synthesis in the generic case, permitted to obtain some generalizations of the Gauss-Bonnet theorem (Agrachev et al. 2008).

Some of the synthesis mentioned above permitted to obtain important results for the theory of hypoelliptic operators (Hormander 1967). Moreover, they permitted to clarify the relation between small-time heat kernel asymptotics and the properties of the value function for the problem (9). See for instance Barilari et al. (2012) and references therein.

Connections with the Stabilization Problem

Consider now the control system $\dot{q}(t) = f(q(t), u(t))$, under the hypothesis (H). Fix $q_0 \in M$ and assume that there exists $u_0 \in U$ such that $f(q_0, u_0) = 0$. A stabilization problem can be stated as follows:

(**P**): For every $\bar{q} \in M$, find a trajectory of the control system $\dot{q}(t) = f(q(t), u(t))$, (under hypothesis (H)) with boundary conditions $q(0) = \bar{q}$, $q(T) = q_0$. (Here T could be required to be finite or not, depending on the problem.)

An elegant way of giving a solution to the problem (**P**) is to give a stabilizing feedback, namely

a function $K(q)$ such that for every $\bar{q} \in M$ the solution of

$$\dot{q}(t) = f(q(t), K(t)) \qquad (10)$$

with initial condition $q(0) = \bar{q}$ steers $\bar{q}$ to q_0.

It is well known that in general it is not possible to give the solution to (**P**) in feedback form. Indeed there may be topological constraints (in the sense of Brockett, see for instance Brockett (1983)) that prevent such a feedback to be continuous. Hence, in general, one cannot guarantee existence and uniqueness of classical or Caratheodory solutions to the ODE (10). This problem attracted a lot of attention since the pioneering work of Brockett and several approaches have been proposed: e.g., via generalized concept of solutions, patchy feedback, time varying feedback etc. (see for instance Clarke et al. 1997; Ancona and Bressan 1999; Coron 1992).

Sometimes one considers an "optimal control" variant of the problem (**P**):

(**Po**): For every $\bar{q} \in M$, find the trajectory of the control system $\dot{q}(t) = f(q(t), u(t))$, (under hypothesis (H)) minimizing the cost $\int_0^T L(q(t), u(t)) \, dt$ (here T can be fixed or free), with boundary conditions $q(0) = \bar{q}, \ q(T) = q_0$.

The cost can be an additional constraint given by the problem, or can be added artificially to have a method and a good concept of solution to solve problem (**P**). Indeed, a way of giving the solution to problem (**Po**) (and hence to (**P**)) is to find the optimal synthesis starting from q_0 for the problem

(**–Po**): for every $\bar{q} \in M$, solve

$$\begin{cases} \dot{q} = -f(q, u), \ u \in U \\ \int_0^T L(q(t), u(t)) \, dt \to \min \\ q(0) = q_0, q(T) = \bar{q}, \end{cases}$$

and then to reverse the time. In other words if $\gamma : [0, T] \to M$ is the solution of (**–Po**) steering q_0 in $\bar{q}$, then $\gamma(T - t)$ is the solution to (**Po**) steering $\bar{q}$ in q_0. This type of solution to problem (**Po**) is called an "optimal stabilizing synthesis".

Extracting a Feedback from an Optimal Synthesis

It is interesting to see what happens if one tries to extract a feedback from an optimal stabilizing synthesis.

If each optimal trajectory of the optimal synthesis corresponds to a regular enough control (e.g., smooth or piecewise) the feedback corresponding to the optimal synthesis can be defined easily in the following way: if $(\gamma(.), u(.))$ defined in $[0, T]$ is a pair trajectory-control of the optimal synthesis, then $K(\gamma(t)) = u(t)$ for every $t \in [0, T]$.

However, as already mentioned, in most of the situations $K(q)$ is not continuous. (Notice that even in the case in which all trajectories of the optimal synthesis are smooth it may happen that $K(q)$ is not continuous.) Hence, in general, one cannot guarantee existence and uniqueness of classical or Caratheodory solutions to the ODE (10).

One could think of enlarging the concept of the solution of (10) by using Filippov, Krasowski, or CLSS (Clarke et al. 1997) solutions (see for instance Marigo and Piccoli 2002, Piccoli and Sussmann 2000 and references therein). However none of these types of solutions are adapted to give the solution of an optimal stabilization problem in feedback form. To fix the ideas, let us consider the case of Filippov solutions. In Piccoli and Sussmann (2000) the authors build examples of optimal synthesis for which the corresponding feedbacks generate solutions that are either Filippov but nonoptimal or optimal but not Filippov. The same can be done with the other types of solutions mentioned above. Also, it is possible to build an example showing an optimal stabilizing synthesis for which the corresponding feedback generates non optimal trajectories even in classical sense. This is presented in the next section.

Hence, at the moment an optimal stabilizing synthesis remains the only possible concept of solution for an optimal stabilizing problem.

Synthesis Theory in Optimal Control, Fig. 1 An optimal stabilizing synthesis for which the corresponding feedback generates nonoptimal trajectories
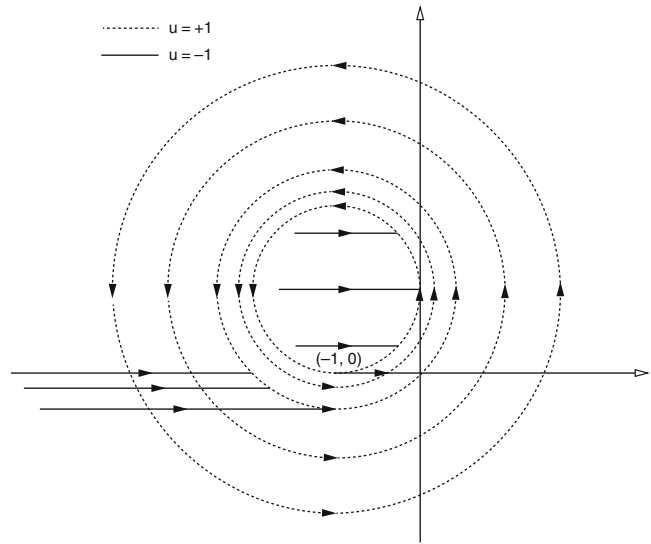
An Example of a Time-Optimal Synthesis Whose Feedback Generates Nonoptimal Trajectories

We present an example exhibiting the phenomenon of nonuniqueness of trajectories for the closed-loop equation arising from the feedback extracted from an optimal synthesis. In particular the optimal feedback admits nonoptimal (classical) solutions. This well illustrates the importance of using the synthesis as concept of solution for an optimal stabilization problem.

Consider the planar system:

$$\dot{q} = F(q) + u\,G(q), \qquad |u| \le 1,$$

where $q = (x, y)$ and:

$$F(q) = \begin{pmatrix} 1 - \frac{y}{2} \\ \frac{x+1}{2} \end{pmatrix}, \qquad G(q) = \begin{pmatrix} -\frac{y}{2} \\ \frac{x+1}{2} \end{pmatrix},$$

and the target is the origin.

The trajectories corresponding to the constant control equal to -1 are straight horizontal lines going from left to right, while those corresponding to $+1$ are circles centered at the point $(-1, 1)$, running counterclockwise. The optimal synthesis is described in Fig. 1. For a proof of optimality see Piccoli and Sussmann (2000).

Starting from the point $(-1, 0)$, we have an infinite number of classical solutions to the discontinuous optimal feedback. Indeed at that point we have $F + G = F - G$, so given any natural number n, the trajectory running n times on the circle centered at $(-1, 1)$ and then going to the origin with control -1 is a classical solution to the discontinuous optimal feeback. However, only the one corresponding to $n = 0$ is optimal.

About other concepts of solutions starting from $(-1, 0)$, one can prove the following. Krasowski or CLSS include classical solutions (and hence produce many nonoptimal trajectories). There is only one Filippov solution, that is the one that rotates indefinitely on the circle and never goes to the origin. This trajectory is not a solution to the stabilization problem since it does not reach the target.

Cross-References

▶ Optimal Control and Mechanics
▶ Sub-Riemannian Optimization

Bibliography

Agrachev A (1996) Exponential mappings for contact sub-Riemannian structures. J Dyn Control Syst 2(3):321–358

Agrachev AA, Sachkov YuL (2004) Control theory from the geometric viewpoint. Encyclopedia of mathematical sciences, vol 87. Springer, Berlin/New York

Agrachev A, Boscain U, Sigalotti M (2008) A Gauss-Bonnet-like formula on two-dimensional almost-Riemannian manifolds. Discret Contin Dyn Syst A 20:801–822

Ancona F, Bressan A (1999) Patchy vector fields and asymptotic stabilization. ESAIM Control Optim Calc Var 4:445–471

Barilari D, Boscain U, Neel RW (2012) Small time heat asymptotics at the sub-Riemannian cut locus. J Differ Geom 92(3):373–416

Beals R, Gaveau B, Greiner P (1996) The Green function of model step two hypoelliptic operators and the analysis of certain tangential Cauchy Riemann complexes. Adv Math 121(2):288–345

Bellaiche A (1996) The tangent space in sub-Riemannian geometry. In: Bellaiche A, Risler J-J (eds) Sub-Riemannian geometry. Progress in mathematics, vol 144. Birkhuser, Basel, pp 1–78

Bloch A (2003) Nonholonomic mechanics and control. Interdisciplinary applied mathematics, vol 24. Springer, New York

Boltyanskii V (1966) Sufficient condition for optimality and the justification of the dynamic programming principle. SIAM J Control Optim 4:326–361

Boscain U, Piccoli B (2004) Optimal synthesis for control systems on 2-D manifolds. SMAI, vol 43. Springer, Berlin/New York

Boscain U, Rossi F (2008) Invariant Carnot-Caratheodory metrics on S^3, $SO(3)$, $SL(2)$ and Lens Spaces. SIAM J Control Optim 47:1851–1878

Boscain U, Duplaix J, Gauthier JP, Rossi F (2012) Anthropomorphic image reconstruction via hypoelliptic diffusion. SIAM J Control Optim 50(3):1309–1336

Breuillard E, Le Donne E (2012) On the rate of convergence to the asymptotic cone for nilpotent groups and subfinsler geometry. PNAS. doi:10.1073/pnas.1203854109

Bressan A (1985) A high order test for optimality of bang-bang controls. SIAM J Control Optim 23(1):38–48

Bressan A, Piccoli B (1998) A generic classification of time optimal planar stabilizing feedbacks. SIAM J Control Optim 36(1):12–32

Bressan A, Piccoli B (2007) Introduction to the mathematical theory of control. AIMS series on applied mathematics, vol 2. American Institute of Mathematical Sciences, Springfield

Brockett R (1982) Control theory and singular Riemannian geometry. In: New directions in applied mathematics (Cleveland, 1980). Springer, New York/Berlin, pp 11–27

Brockett R (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Suss-mann HJ (eds) Differential geometric control theory. Birkhäuser, Boston, pp 181–191

Brunovsky P (1978) Every normal linear system has a regular time-optimal synthesis. Math Slovaca 28:81–100

Brunovsky P (1980) Existence of regular syntheses for general problems. J Differ Equ 38:317–343

Cesari L (1983) Optimization-theory and applications: problems with ordinary differential equations. Springer, New York

Clarke F, Ledyaev Yu, Subbotin A, Sontag E (1997) Asymptotic controllability implies feedback stabilization. IEEE Trans Autom Control 42:1394–1407

Charlot G (2002) Quasi-contact S-R metrics: normal form in $\mathbb{R}^{2n}$, wave front and caustic in $\mathbb{R}^4$. Acta Appl Math 74(3):217–263

Coron JM (1992) Global asymptotic stabilization for controllable systems without drift. Math Control Signals Syst 5:295–312

Dubins LE (1957) On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal position and tangents. Am J Math 79:497–516

El-Alaoui El-H Ch, Gauthier J-P, Kupka I (1996) Small sub-Riemannian balls on $\mathbb{R}^3$. J Dyn Control Sys 2(3):359–421

Gaveau B (1977) Principe de moindre action, propagation de la chaleur et estimées sous elliptiques sur certains groupes nilpotents. Acta Math 139(1–2):95–153

Gershkovich V, Vershik A (1988) Nonholonomic manifolds and nilpotent analysis. J Geom Phys 5:407–452

Gromov M (1981) Groups of polynomial growth and expanding maps. Inst Hautes Ètudes Sci Publ Math 53:53–73

Hormander L (1967) Hypoelliptic second order differential equations. Acta Math 119:147–171

Krener AJ (1977) The high order maximal principle and its application to singular extremals. SIAM J Control Optim 15(2):256–293

Marigo A, Piccoli B (2002) Regular syntheses and solutions to discontinuous ODEs. ESAIM Control Optim Calc Var 7:291–308

Montgomery R (2002) A tour of subriemannian geometries, their geodesics and applications. Mathematical surveys and monographs, vol 91. American Mathematical Society, Providence

Petitot J (2008) Neurogéométrie de la vision, Modèles mathématiques et physiques des architectures fonctionnelles. Les Éditions de l' École Polytechnique

Piccoli B (1996) Classifications of generic singularities for the planar time-optimal synthesis. SIAM J Control Optim 34(6):1914–1946

Piccoli B, Sussmann HJ (2000) Regular synthesis and sufficiency conditions for optimality. SIAM J Control Optim 39(2):359–410

Pontryagin LS et al (1961) The mathematical theory of optimal processes. Wiley, New York

Reeds JA, Shepp LA (1990) Optimal Path for a car that goes both forwards and backwards. Pac J Math 145:367–393

Sachkov Yu (2011) Cut locus and optimal synthesis in the sub-Riemannian problem on the group of motions of a plane. ESAIM COCV 17:293–321

Sigalotti M, Chitour Y (2006) Dubins' problem on surfaces II: nonpositive curvature. SIAM J Control Optim 45:457–482

Soueres P, Laumond JP (1996) Shortest paths synthesis for a car-like robot. IEEE Trans Autom Control 41(5):672–688

Sussmann HJ (1979) Subanalytic sets and feedback control. J Differ Equ 31(1):31–52

Sussmann HJ (1980) Analytic stratifications and control theory. In: Proceedings of the international congress of mathematicians (Helsinki, 1978), Academia Scientiarum Fennica, Helsinki, pp 865–871

Sussmann HJ (1986) Envelopes, conjugate points, and optimal bang-bang extremals. In: Algebraic and geometric methods in nonlinear control theory. Mathematics and its applications, vol 29. Reidel, Dordrecht, pp 325–346

Sussmann HJ (1989) Envelopes, higher-order optimality conditions and Lie Brackets. In: Proceedings of the 1989 IEEE conference on decision and control, Tampa, FL, USA

Vinter R (2010) Optimal control. Birkhäuser, Basel/Boston

Synthetic Biology

Domitilla Del Vecchio[1] and Richard M. Murray[2]
[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
[2]Control and Dynamical Systems, Caltech, Pasadena, CA, USA

Abstract

The past decade has seen tremendous advances in DNA recombination and measurement techniques. These advances have reached a point in which de novo creation of biomolecular circuits that accomplish new functions is now possible, leading to the birth of a new field called synthetic biology. Sophisticated functions that are highly sought in synthetic biology range from recognizing and killing cancer cells, to neutralizing radioactive waste, to efficiently transforming feedstock into fuel, to control the differentiation of tissue cells. To reach these objectives, however, there are a number of open problems that the field has to overcome. Many of these problems require a system-level understanding of the dynamical and robustness properties of interacting systems, and hence, the field of control and dynamical systems theory may highly contribute. In this entry, we review the basic technology employed in synthetic biology and a number of simple modules and complex systems created using this technology and discuss key system-level problems along with challenging research questions for the field of control theory.

Keywords

Biomolecular systems; Gene expression; Robustness; Modularity

Introduction to Synthetic Biology

Synthetic biology is an emerging engineering discipline in which the biochemical and biophysical principles present in living organisms are used to engineer new systems (Baker et al. 2006). These systems will have the ability of accomplishing a number of remarkable tasks, such as turning waste into energy sources, neutralizing radioactive waste, detecting environmental pathogens, or recognizing cancer cells with the aim of targeting them for deletion. While synthetic biology can be employed to create new functionalities, it can also enable the understanding of fundamental design principles of living systems. In fact, implementing a circuit with a prescribed behavior provides a powerful means to test hypotheses regarding the underlying biological mechanisms.

The functions of living organisms are controlled by biomolecular circuits, in which proteins and genes interact with each other through activation and repression interactions forming complex networks. A common signal carrier is the concentration of the active form of a protein, which can be controlled through a number of mechanisms, including gene expression regulation and post-translational

S

modification. Through the process of gene expression, proteins are produced by their corresponding genes, whose production rates can be activated or repressed by other proteins (transcription factors). Once the proteins are produced, they can be activated or inhibited, by other proteins or smaller molecules, through post-translation modification processes including covalent modification, such as phosphorylation, and allosteric modification (Alon 2007). We next describe some salient aspects of gene expression focusing, for simplicity, on prokaryotic systems.

A gene is a piece of DNA whose expression rate can often be controlled by a DNA sequence upstream of the gene itself, called promoter. The promoter contains the binding regions for the RNA polymerase, an enzyme that transcribes the gene into a messenger RNA molecule, which is then translated into protein by the ribosomes. The promoter also contains operator sites, which are binding regions where other proteins, called transcription factors, can bind. If these proteins are activators, they will help the RNA polymerase in binding the promoter to start transcription. By contrast, if these proteins are repressors, they will prevent the RNA polymerase from binding the promoter. These activation and repression inter-actions are highly nonlinear and often stochastic; therefore, the most commonly used modeling frameworks include systems of nonlinear ordinary differential equations, stochastic differential equations, or the chemical master equation (Gillespie 1977, 2000).

The basic technique for constructing synthetic circuits is that of assembling, through the process of cloning, DNA sequences with prescribed combinations of promoters and genes such that a desired network of activation and repression interaction is created. For example, if we would like to create an inverter where protein A represses protein B, we can simply place the gene of B under the control of a promoter repressed by protein A. Currently, there is a library of parts that one can use to assemble a desired circuit this way. The set of parts includes promoters, gene coding sequences, terminators, and ribosome binding sites. Terminators are DNA sequences placed at the end of a gene to make the RNA polymerase terminate transcription, while ribosome binding sites are DNA sequences placed at the beginning of a gene, which establish the rate at which ribosomes will bind to the mRNA, determining the overall translation rate (Endy 2005). An area of intense research is the expansion of the library by creating mutations of existing parts or by assembling new ones.

Once a DNA sequence is created that encodes the desired circuit, it is inserted in a living cell either on the chromosome itself or on DNA plasmids. When the circuit is inserted in the chromosome, it will be in one copy, while when it is inserted in DNA plasmids, it will be in as many copies as the plasmid copy number. Plasmid copy number can vary from low copy (5–10 copies), to medium copy (20 copies), to high copy (about 100 copies). Once in the cell, the circuit will have the required resources to function, including RNA polymerase, ribosomes, amino acids, and ATP (the cell energy currency). In this sense, the cell can be viewed as a chassis for the synthetic circuits. The operation of the circuit can then be observed by monitoring the concentration of reporters, that is, of proteins that are easy to detect and quantify. These include fluorescent proteins, that is, proteins that exhibit bright fluorescence when exposed to light of a specific wave length. Examples include the green, red, blue, and yellow fluorescent proteins. These fluorescent proteins are mainly employed in two different ways to measure the amount of a protein of interest. One can fuse the gene of the fluorescent protein with the gene expressing the protein of interest. Alternatively, one can use the protein of interest as a transcription factor of the fluorescent protein. In both cases, the concentration of the fluorescent protein will provide an indirect measurement of the concentration of the protein of interest.

It is also possible to apply external inputs to a circuit to control the activity of transcription factors. This is accomplished through the use of inducers, which are small signaling molecules that can be injected in the cell culture and enter the cell wall. These inducers bind specific transcription factors and either activate them,

allowing the transcription factor to bind the promoter operator sites, or inhibit them, reducing the transcription factor's ability to bind the promoter operator sites.

Examples of Synthetic Biology Modules

A number of modules comprising two or three genes have been fabricated in the earlier days of synthetic biology (Atkinson et al. 2003; Becskei and Serrano 2000; Elowitz and Leibler 2000; Gardner et al. 2000; Stricker et al. 2008). We can group them into oscillators (Atkinson et al. 2003; Elowitz and Leibler 2000; Stricker et al. 2008), mono-stable systems (Becskei and Serrano 2000), and bistable systems called toggle switches (Gardner et al. 2000). More recently, feedforward loops have also been fabricated (Bleris et al. 2011).

Oscillators. The creation of circuits whose protein concentrations oscillate periodically in time has been a major focus. In fact, the ability of creating an oscillator has the potential of shedding light into the mechanisms at the basis of natural clocks, such as circadian rhythms and the cell cycle. Oscillator designs can be divided into two types: loop oscillators (Elowitz and Leibler 2000), in which repression/activation interactions occur in a loop topology, or oscillators based on the interplay between an autocatalytic loop and negative feedback (Atkinson et al. 2003; Stricker et al. 2008) (see Fig. 1).

The design requirements of synthetic circuits are usually explored through models of varying detail, starting with the use of low-dimensional "toy models," which are composed of a set of nonlinear ordinary differential equations describing the rate of change of the circuit's proteins. These models allow application of a number of tools from dynamical systems theory to infer parameter or structural requirements for a desired behavior. After toy models are analyzed, larger-scale mechanistic models are constructed, which include all the intermediate species taking part in the biochemical reactions. These models can be
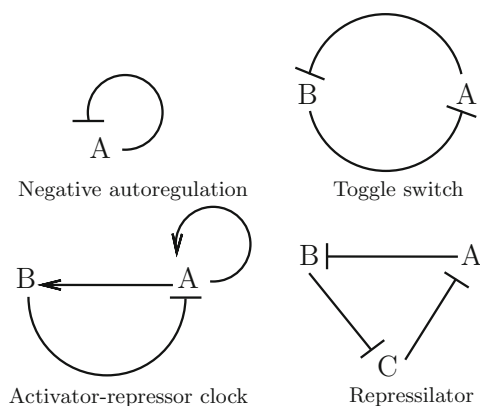
Synthetic Biology, Fig. 1 Early gene circuits that have been fabricated in bacteria *E. coli*: the negatively autoregulated gene (Becskei and Serrano 2000), the toggle switch (Gardner et al. 2000), the activator-repressor clock (Atkinson et al. 2003), and the repressilator (Elowitz and Leibler 2000)

either deterministic or stochastic. Simulation is usually required for the study of these more complicated models, and the Gillespie algorithm is often employed for stochastic simulations (Gillespie 1977).

As an example of a toy model and related analysis, consider the activator-repressor clock of Atkinson et al. (2003) shown in Fig. 1. This oscillator is composed of an activator A activating itself and a repressor B, which, in turn, represses the activator A. Both activation and repression occur through transcription regulation. Denoting in italics the concentration of species, a toy model of this clock can be written as

$$
\begin{aligned}
\dot{A} &= \frac{\beta_A (A/K_a)^n + \beta_{0,A}}{1 + (A/K_a)^n + (B/K_b)^m} - \gamma_A A, \\
\dot{B} &= \frac{\beta_B (A/K_a)^n + \beta_{0,B}}{1 + (A/K_a)^n} - \gamma_B B,
\end{aligned}
\tag{1}
$$

in which γ_A and γ_B represent protein decay (due to dilution and/or degradation). The functions $(\beta_A (A/K_a)^n + \beta_{0,A})/(1 + (A/K_a)^n + (B/K_b)^m)$ and $(\beta_B (A/K_a)^n + \beta_{0,B})/(1 + (A/K_a)^n)$ are called Hill functions and are the most commonly used models for transcription regulation (Alon 2007). The first Hill function in system (1) increases with A and decreases with B, while
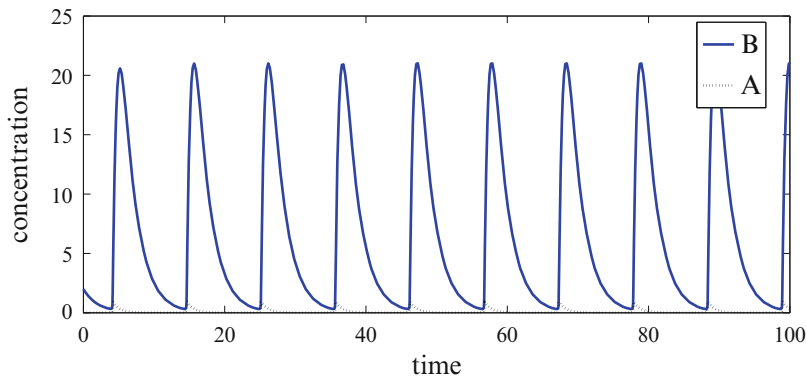
Synthetic Biology, Fig. 2 Activator-repressor clock time trajectory

the second one increases with *A*, as expected since A is an activator and B is a repressor. The key mechanism by which this system displays sustained oscillations is a supercritical Hopf bifurcation with bifurcation parameter the relative timescale of the activator dynamics with respect to the repressor dynamics (Del Vecchio 2007). Specifically, as the activator dynamics become faster than the repressor dynamics, the system goes through a supercritical Hopf bifurcation and a stable periodic orbit appears (Fig. 2).

Mono-stable systems. The mono-stable system engineered through negative autoregulation was fabricated with the aim of understanding the role of negative feedback in attenuating biological noise. The results of Becskei and Serrano (2000) clearly showed that negative autoregulation can reduce intrinsic noise. Furthermore, the results of Austin et al. (2005) demonstrated that while low frequency noise is attenuated, noise at high frequency can be amplified by negative autoregulation in accordance with Bode's integral formula (Åström and Murray 2008).

Bistable systems. The toggle switch of Gardner et al. (2000) was the first bistable system constructed. It constitutes the simplest circuit with memory, in which the state of the system can be switched from one equilibrium (low, high) to the other (high, low) by external inputs. Once the system state is switched to one of these two equilibria, it will stay there unless another external perturbation is applied.

Feedforward loops. While the early circuits described so far were fabricated mainly to investigate design principles for limit cycles and for robustness, many more circuits after those have been fabricated with the aim of solving concrete engineering problems. As an example, the incoherent feedforward circuit of Bleris et al. (2011) was fabricated in bacteria *E. coli* with the aim of making protein production independent of DNA plasmid copy number. In fact, DNA copy number fluctuates stochastically with possibly large deviations from the nominal value. As a consequence, the concentration of proteins expressed from genes residing on a plasmid also fluctuates stochastically. In order to make protein concentration independent of an unknown DNA copy number, one could leverage principles for disturbance rejection such as integral control. While an explicit integral control action is particularly hard to implement through biological parts, incoherent feedforward loops are easier to implement and can accomplish the same disturbance rejection task. In these loops, the disturbance input affects the output through two branches, one in which the disturbance activates the output and a longer one in which the disturbance represses the output (Alon 2007). If these two branches are appropriately balanced, the steady-state value of the output will be practically independent of the disturbance input, leading to disturbance rejection to constant or slowly changing disturbances.

From Modules to Systems

One approach to creating systems that can accomplish sophisticated tasks is to assemble together simpler modules, such as those described in the previous section (Purnick and Weiss 2009). For example, the artificial tissue homeostasis circuit proposed by Miller et al. (2012) is composed of several interconnected modules, including an activator-repressor clock, a toggle switch, a couple of inverters, and an "and" gate. Control of tissue homeostasis refers to the ability of regulating a cell type to a constant level in a multicellular community. This ability is central in several diseases such as cancer and diabetes, in which tissue homeostasis is misregulated. The design proposed by Miller et al. (2012) illustrates how a synthetic biological circuit can be modularly created to accomplish this complicated regulation function.

Layered logic gates are often necessary in order to integrate multiple signals. Moon et al. (2012) have constructed an "and" gate that integrates more than two signals by cascading pairs of "and" gates. Of course, problems of latency become more relevant as the number of layers increases and methods to mitigate these effects are being developed.

An application that requires the integration of multiple signals is the cell-type classifier of Xie et al. (2011). Here, a synthetic gene circuit is created that integrates sensory information from a number of molecular markers to determine whether a cell is in a specific state, that is, cancer, and, in such a case, produces a protein output triggering cell death. The design of this circuit is based on the composition of three key modules. Specifically, a double inversion module senses high levels of a molecular marker, a single inversion module senses low levels of a molecular marker, and a logical "and" module finally integrates the outputs of the other two modules to produce the output protein.

Finally, biofuels are another high-impact application of synthetic biology (Peralta-Yahya et al. 2012). Metabolic engineering has been employed for a long time in order to engineer microbes to produce advanced biofuels with similar properties to petroleum-based fuels. One challenge in using microbes (or other living organisms) to convert feedstock into biofuel is that of overcoming the endogenous cell regulation to achieve sufficiently high yields such that advanced biofuels are economically advantageous. Specifically, engineered pathways are optimized on the basis of nominal operating conditions, but these conditions often change when microbes are in bioreactors. To mitigate this problem, synthetic gene circuits have been designed to sense the metabolic status of the host and regulate key points in the metabolic pathway to optimize yield (Zhang et al. 2012).

Main System-Level Challenges to Design

One major challenge in synthetic biology is the ability of going from simple modules to larger sophisticated systems (Purnick and Weiss 2009). Problems in advancing in this direction can be divided into two categories: "hardware" problems and system-level problems. Hardware problems include issues such as the availability of enough orthogonal parts to allow scaling up the size of synthetic circuits. We do not expand on this here and instead focus on system-level problems. These include issues such as *context dependence* (Cardinale and Arkin 2012), that is, the fact that modules behave in a poorly predictable way once interacting together in the cell environment. This is a major obstacle to creating larger circuits that behave predictably.

Problems of context dependence can be further divided into three qualitatively different types: (a) inter-modular interactions, (b) interactions of synthetic circuits with the cell machinery, (c) perturbations in the external environment. We analyze each of them separately.

(a) When modules are connected to each other to create larger systems, a protein in an upstream module is used as an "input" to

a downstream module. This fact creates a "loading" on the upstream system due to the fact that the output protein cannot take part in the upstream module reactions whenever it is taking part in the downstream module reactions. As a consequence, the behavior of the upstream system changes compared to when the system functions in isolation (Del Vecchio et al. 2008; Saez-Rodriguez et al. 2004). These loading effects have been called retroactivity to extend the notion of loading and impedance to biomolecular systems. Accordingly, solutions to mitigate this problem are being investigated (Franco et al. 2011; Jayanthi and Del Vecchio 2011; Mishra et al. 2013).

(b) Ideally, the cell should function as a "chassis" for synthetic biology circuits. In practice, this is not the case because the endogenous circuitry interacts with synthetic circuits even when parts that are orthogonal to the endogenous systems are employed. A major example of this interaction is the depletion of cellular resources, such as ATP, RNA polymerase, and ribosomes, which are required for the operation of synthetic circuits. This depletion reduces cell fitness, with deleterious consequences also for synthetic circuits, a phenomenon called "metabolic burden" (Bentley et al. 1990). A more subtle phenomenon than purely reducing cell fitness is that synthetic circuits compete with each other for the same resources. This fact creates implicit and unwanted coupling among circuits with unpredictable consequences. Approaches to mitigate these problems are under investigation. One direction is the use of orthogonal RNA polymerase and ribosomes (Wenlin and Chin 2009; Rackham and Chin 2005). A completely different, but complementary, direction is that of establishing implementable design principles that allow circuits to function robustly despite fluctuations in the resources they use.

(c) The external environment where a cell operates has a number of physical attributes, which may also be subject to perturba-

tions. These physical attributes include temperature, acidity, nutrients' level, etc. Perturbations in these attributes often lead to poor cell fitness or to nonstandard growth conditions, ultimately leading to synthetic circuits malfunctions.

Summary and Future Directions

The future of synthetic biology highly depends on the ability of scaling up the complexity of design to create more sophisticated functions. While a number of issues, such as the availability of enough orthogonal parts, can be successfully addressed by (nontrivial) fabrication of new parts, issues such as context dependence require a system-level dynamic understanding of circuits and their interactions. Here is where control and dynamical systems theory could greatly contribute. Control theory has proven critical to reason about and engineer robustness in a number of concrete applications including aerospace and automotive systems, robotics and intelligent machines, manufacturing chains, electrical, power, and information networks. Similarly, control theory could enable the understanding of principles that ensure robust behavior of synthetic circuits once interacting with each other in the cell environment, leading to the ultimate progress of synthetic biology.

A number of challenges need to be addressed for the successful application of control and dynamical systems theory to synthetic biology. The behavior of synthetic circuits is highly nonlinear and, as a consequence, control theoretic tools designed for understanding robustness in linear systems are not directly applicable. Understanding how to exploit the rich structure of biomolecular circuits to quantitatively reason about robustness to interconnections, competition for shared resources, and fluctuations of temperature and nutrients is likely to have a major impact. Even with this understanding, however, the question of how to implement robust designs with the currently available biomolecular mechanisms must be addressed. Stochasticity is another major problem since the behavior of synthetic circuits is intrin-

sically noisy. Unfortunately, the availability of analytical tools that allow quantification of how perturbations and uncertainty propagate through a nonlinear stochastic system is still limited, and designers often resort to stochastic simulation. Finally, the values of the salient parameters of the available parts are poorly known. Physical attributes such as binding affinities, ribosome binding site strengths, promoter strengths, etc. are only known within very coarse bounds. These bounds are also usually determined based on a specific organism and in specific growth conditions, which may be different from the ones in which the circuit is ultimately running. Hence, a central question is how to design and implement a system such that the prescribed behavior is robust to all sources of perturbations described above within a large range of possible parameter values.

Cross-References

▶ Deterministic Description of Biochemical Networks

▶ Identification and Control of Cell Populations

▶ Robustness Analysis of Biological Models

▶ Stochastic Description of Biochemical Networks

Bibliography

Alon U (2007) An introduction to systems biology. Design principles of biological circuits. Chapman-Hall, Boca Raton

Åström KJ, Murray RM (2008) Feedback systems. Princeton University Press, Princeton

Atkinson MR, Savageau MA, Meyers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *Escherichia coli*. Cell 113:597–607

Austin DW, Allen MS, McCollum JM, Dar RD, Wilgus JR, Sayler GS, Samatova NF, Cox CD, Simpson ML (2005) Gene network shaping of inherent noise spectra. Nature 439:608–611

Baker D, Church G, Collins J, Endy D, Jacobson J, Keasling J, Modrich P, Smolke C, Weiss R (2006) Engineering life: building a FAB for biology. Sci Am 294:44–51

Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation. Nature 405:590–593

Bentley WE, Mirjalili N, Andersen DC, Davis RH, Kompala DS (1990) Plasmid-encoded protein: the principal factor in the "metabolic burden" associated with recombinant bacteria. Biotechnol Bioeng 35(7):668–681

Bleris L, Xie Z, Glass D, Adadey A, Sontag E, Benenson Y (2011) Synthetic incoherent feedforward circuits show adaptation to the amount of their genetic template. Mol Syst Biol 7:519

Cardinale S, Arkin AP (2012) Contextualizing context for synthetic biology – identifying causes of failure of synthetic biological systems. Biotechnol J 7:856–866

Del Vecchio D (2007) Design and analysis of an activator-repressor clock in *E. coli*. In: Proceedings of the American control conference, New York, pp 1589–1594

Del Vecchio D, Ninfa AJ, Sontag ED (2008) Modular cell biology: retroactivity and insulation. Mol Syst Biol 4:161

Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403:339–342

Endy D (2005) Foundations for engineering biology. Nature 438(24):449–452

Franco E, Friedrichs E, Kim J, Jungmann R, Murray R, Winfree E, Simmel FC (2011) Timing molecular motion and production with a synthetic transcriptional clock. Proc Natl Acad Sci. doi:10.1073/pnas.1100060108

Gardner TS, Cantor CR, Collins JJ (2000) Construction of the genetic toggle switch in *Escherichia Coli*. Nature 403:339–342

Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. J Phys Chem 81:2340–2361

Gillespie DT (2000) The chemical Langevin equation. J Chem Phys 113:297–306

Jayanthi S, Del Vecchio D (2011) Retroactivity attenuation in bio-molecular systems based on timescale separation. IEEE Trans Autom Control 56:748–761

Miller M, Hafner M, Sontag E, Davidsohn N, Subramanian S, Purnick P, Lauffenburger D, Weiss R (2012) Modular design of artificial tissue homeostasis: robust control through synthetic cellular heterogeneity. PLoS Comput Biol 8:e1002579

Mishra D, Rivera-Ortiz P, Del Vecchio D, Weiss R (2013) A load driver device for engineering modularity in biological networks. Nat Biotechnol (Under review, accepted and to appear)

Moon TS, Lou C, Tamsir A, Stanton BC, Voigt CA (2012) Genetic programs constructed from layered logic gates in single cells. Nature 491:249–253

Peralta-Yahya PP, Zhang F, del Cardayre SB, Keasling JD (2012) Microbial engineering for the production of advanced biofuels. Nature 488:320–328

Purnick P, Weiss R (2009) The second wave of synthetic biology: from modules to systems. Nat Rev Mol cell Biol 10:410–422

Rackham O, Chin JW (2005) A network of orthogonal ribosome-mRNA pairs. Nat Chem Biol 1(3):159–166

Saez-Rodriguez J, Kremling A, Conzelmann H, Bettenbrock K, Gilles ED (2004) Modular analysis of

signal transduction networks. IEEE Control Syst Mag 24(4):35–52

Stricker J, Cookson S, Bennett MR, Mather WH, Tsimring LS, Hasty J (2008) A fast, robust and tunable synthetic gene oscillator. Nature 456:516–519

Wenlin A, Chin JW (2009) Synthesis of orthogonal transcription translation networks. Proc Natl Acad Sci 106(21):8477–8482

Xie Z, Wroblewska L, Prochazka L, Weiss R, Benenson K (2011) Multi-input rnai-based logic circuit for identification of specific cancer cells. Science 333:1307–1311

Zhang F, Carothers JM, Keasling JD (2012) Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. Nat Biotechnol 30:354–359

System Identification Software

Brett Ninness
School of Electrical and Computer Engineering, University of Newcastle, Newcastle, Australia

Abstract

This contribution discusses various aspects important to software for system identification. Essential functionality for existing practice and the algorithmic fundamentals this relies on are considered together with a brief discussion of additional commonly useful support tools. Since software is intimately tied to the hardware that it runs on, a discussion on this topic follows with an emphasis on considering how future system identification software developments might best align with clear current and future trends in computer architecture developments.

Keywords

System identification; Computer-aided design; Parameter estimation; Software

Introduction

Fundamental to the practice of system identification is the employment of appropriate software to compute system estimates and evaluate their properties. One option is for the user to code the necessary routines themselves in their computer language of choice. For simple situations, such as least-squares estimation with a linearly parametrized model, this approach is feasible.

However, it quickly becomes onerous and time consuming as one moves even slightly beyond this simple example. In response to this, researchers have developed a number of software packages designed to accommodate classes of data formats, model structures, and estimation methods.

The purpose of this contribution is to profile the support that available system identification software provides, the underlying foundations on which this software depends, and the future capabilities that may be expected due to trends in desktop and portable computer capacity.

The material to follow depends on explanations, definitions, and background presented in ▶ System Identification: An Overview, by Ljung, which should be read in conjunction with this contribution.

Essential Functionality

The essence of system identification software packages is that they implement an identification method $\mathcal{I}$ as defined in ▶ System Identification: An Overview.

Typically, this involves taking a model structure specification $\mathcal{M}(\theta)$ together with N observed data points Z_N and translating that to a cost function $V_N(\theta)$ for which a minimizer

$$\hat{\theta} \triangleq \underset{\theta \in D_{\mathcal{M}}}{\arg\min} \, V_N(\theta) \qquad (1)$$

is then computed in order to deliver a system estimate $\mathcal{M}(\hat{\theta})$.

While the details of these fundamental operations vary according to the chosen model structure and method, there are some shared aspects. To pick a starting point, subspace-based estimation methods (▶ Subspace Techniques in System

Identification) have been one of the most significant developments in the near history of system identification, and they fundamentally involve a first stage of setting up and solving the optimization problem

$$\hat{\beta} = \arg\min_{\beta} \|Y - \Phi\beta\|_F^2, \qquad (2)$$

where Y, Φ are data-dependent matrices, β is a θ- dependent matrix, and $\|\cdot\|_F$ is the Frobenius norm, which, for an $m \times n$ matrix A, is defined as

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}. \qquad (3)$$

This is a classic least-squares optimization problem, which also arises in other system identification contexts, particularly when the prediction $\hat{y}(t \mid \theta)$ is a linear function of θ.

As is well known Golub and Loan (1989), the minimizer $\hat{\beta}$ satisfies the "normal equations"

$$(\Phi^T\Phi)\hat{\beta} = \Phi^T Y, \qquad (4)$$

and if $\Phi^T\Phi$ is invertible, this allows for a closed-form solution

$$\hat{\beta} = (\Phi^T\Phi)^{-1}\Phi^T Y. \qquad (5)$$

While formally correct, no system identification software packages would compute $\hat{\beta}$ in this manner since it is computationally inefficient and sensitive to numerical rounding errors.

Drawing on decades of study on this topic in the numerical computations literature (Golub and Loan 1989), system identification software packages rely on the QR factorization

$$\Phi = QR = [Q_1 \mid Q_2]\begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \qquad (6)$$

where Q is square and satisfies $Q^T Q = I$ (the identity matrix) and R contains the upper triangular square and invertible block R_1. This decomposition of Φ allows the normal Eq. (4) to be re-expressed as

$$R_1\hat{\beta} = Q_1^T Y. \qquad (7)$$

Since R_1 is upper triangular, the solution $\hat{\beta}$ may then be found by elementary and numerically robust backward substitution (Golub and Loan 1989).

The importance of efficient and accurate solution of normal equations to any system identification software is not limited to these subspace or linearly parametrized cases. For instance, the very general class of prediction error methods encompassed by the formulation (1) involves a cost $V_N(\theta)$ that depends on the vector

$$E(\theta) \triangleq [\varepsilon(t_1, \theta), \cdots, \varepsilon(t_N, \theta)]^T \qquad (8)$$

of differences between the observed data and the response of a model parametrized by θ. In the case of time-domain data, the elements of (8) are defined by

$$\varepsilon(t, \theta) \triangleq y(t) - \hat{y}(t \mid \theta). \qquad (9)$$

In this general situation, it is most commonly the case that no closed-form solution for the optimization problem (1) exists.

The strategy then taken by most system identification software packages is to employ a gradient-based search for a minimizer. These methods are motivated by the use of a linear approximation of $E(\theta)$ about a current putative minimizer θ_k according to

$$E(\theta) \approx E(\theta_k) + J(\theta_k)(\theta - \theta_k), \qquad (10)$$

where $J(\theta_k)$ denotes the Jacobian matrix

$$J(\theta_k) \triangleq \left.\frac{\partial E(\theta)}{\partial \theta}\right|_{\theta=\theta_k}. \qquad (11)$$

In the very common situation where $V_N(\theta)$ is a quadratic function of $E(\theta)$, this implies the associated approximation

$$V_N(\theta) = \text{Trace}\{E^T(\theta)E(\theta)\}$$
$$= \|E\|_F^2 \approx \|E(\theta_k) + J(\theta_k)(\theta - \theta_k)\|_F^2. \qquad (12)$$

S

Via this reasoning, computation of an appropriate "search direction" $p = \theta - \theta_k$ again involves the efficient solution of a linear least-squares problem of the form (2), namely,

$$p = \arg\min_p \|E(\theta_k) + J(\theta_k)\, p\|_F^2. \qquad (13)$$

More generally, system identification software packages extend this rationale and solve (1) by generating a sequence of iterations $\{\theta_k\}$, which are refined according to

$$\theta_{k+1} = \theta_k + \mu\, p, \qquad (14)$$

where μ is a step length that at each iteration k may be altered until a cost decrease

$$V_N(\theta_{k+1}) < V_N(\theta_k) \qquad (15)$$

is achieved and the search direction p again involves the solution of normal equations

$$\left[J(\theta_k)^T J(\theta_k) + \lambda I \right] p = -J(\theta_k)^T E(\theta_k). \qquad (16)$$

The choice $\lambda > 0$ implies what is called a Levenberg–Marquardt method, while $\lambda = 0$ leads to a so-called Gauss–Newton update strategy, and there are further variants such as "trust region" methods that are typically offered as options.

Via (16) we see that again system identification software comes to fundamentally depend on underpinning numerical linear algebra, in this case, again via the QR decomposition.

Another decomposition, the singular value decomposition (SVD), also has a significant role to play, particularly with respect to subspace-based methods where it is essential to the extraction of an estimated system parametrization $\hat{\theta}$ from $\hat{\beta}$ referred to in (2).

In addition to matrix decompositions, other system identification methods depend on many other even more fundamental linear algebra tools such as basic matrix/vector operations, matrix inversion, and eigen-decomposition. Because of this dependence, most (Ljung 2012; Kollár et al. 2006; Young and Taylor 2012; Garnier et al.

2012; Ninness et al. 2013) but not all (Hjalmarsson and Sjöberg 2012) currently available system identification software packages are built upon the MathWorks MATLAB (originally short for "matrix laboratory") package, which provides an efficient interface to the widely accepted standard numerical linear algebra libraries LAPACK and EISPACK. For example, solving (2) efficiently and robustly via QR decomposition and back-substitution of (7) is achieved transparently using the MATLAB backslash operator with the simple command: `beta = Phi\Y`.

Additional Functionality and the Decision-Making Process

As emphasized in ▶ System Identification: An Overview, the provision of an estimated model is typically an iterative process (illustrated diagrammatically in Fig. 4 of ▶ System Identification: An Overview) of which just one component is the implementation of an identification method $\mathcal{I}$ to deliver a system estimate $\mathcal{M}(\hat{\theta})$.

In addition to this "essential functionality," system identification software must also provide tools and a logistical support for the decision-making process of assessing $\mathcal{M}(\hat{\theta})$ and, based on this, perhaps altering aspects such as the choice of model structure $\mathcal{M}$, the experiment design $\mathcal{X}$, or indeed the identification method $\mathcal{I}$.

To support this, system identification software packages may offer further capabilities such as:

1. Nonparametric estimation methods that deliver estimates of linear system frequency response without involving a parametrized model structure $\mathcal{M}(\theta)$ and hence not involving (1)
2. Data preprocessing tools, such as to remove trends and to frequency selectively prefilter data before use
3. Visualization tools to display and compare the time- and frequency-domain response of estimated models
4. Model validation tools to determine if estimated models can be falsified by observed data

5. Model accuracy measures that deliver statistical confidence bounds on estimated parameters

6. Additional data processing tools such as Kalman filtering and smoothing routines and sequential Monte Carlo (particle filter) routines that are used to compute $V_N(\theta)$ but have many other applications

7. Graphical user interface (GUI) support in order to aid organization of the various aspects of data preprocessing, model structure selection, algorithm selection, estimate computation, model validation, and model visualization

8. The employment of symbolic computation capabilities to aid complex model structure specification and preprocessing for efficient numerical implementation (Hjalmarsson and Sjöberg 2012)

Note that with the exception of this last point (8), the computations associated with this additional functionality again depend fundamentally on efficient numerical linear algebra software.

Computing Platforms

Currently available system identification software packages are designed for standard desktop computing environments, and as such their capabilities are intimately tied to those of the central processing unit (CPU), memory, and other architectural features of this hardware.

For instance, the linear algebra underpinnings just discussed are typically implemented in serially coded form, and hence bus bandwidth, together with memory and CPU speed, will be the fundamental factor affecting software performance. Taking CPU speed as an example, the evolution of clock speed for the very commonly used Intel architecture CPUs is shown as the red curve in Fig. 1 and, as can be seen, has largely plateaued over the last decade after two orders of magnitude growth in the decade preceding it.

As a result, and roughly speaking, system estimates that took a minute to compute in the early 1990s took under a second to compute in the early part of this century, but are essentially no faster to compute now, a further decade later.

As a result, while system identification software has continued to grow in sophistication, in areas that involve high computational burdens, such as estimation of complex and high-dimensional model structures, or the implementation of compute intensive algorithms, the capability of system identification software has been hardware limited for some time.

At the same time, as the blue line in Fig. 1 illustrates, Moore's law continues to hold, and

System Identification Software, Fig. 1 Trends in desktop CPU capacity taking Intel as an example. Serial throughput speeds have long plateaued, but transistor density continues to grow, which delivers growing multiple cores
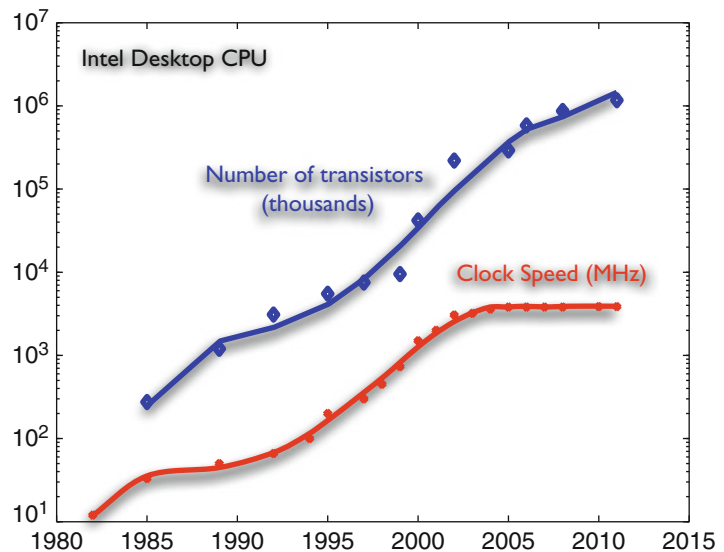
transistor densities continue to increase. While this is delivering no greater serial CPU speed, it is delivering multiple CPU core availability. Future advances in system identification software capability will therefore need to exploit the potential for parallel computation.

Indeed, in current MATLAB, the fundamental numerical linear algebra routines previously mentioned such as QR-based solution of normal equations, eigenvalue, and SVD decompositions will all automatically execute on multiple computational threads on multicore-enabled machines. Expanding this to take advantage of even higher levels of parallelism is the subject of current research.

While these developments will deliver performance enhancements for existing system identification methods, they will also open up the possibility for new tools to be added to system identification software suites.

For example, in addition to the existing subspace, prediction error, and maximum likelihood methods just mentioned, there is another important estimation approach that does not involve the solution of an optimization problem such as (1) or (2) and for which there is always a closed-form expression for the parameter estimate. It is the conditional mean estimate

$$\hat{\theta} = \mathbf{E}\{\theta \mid Y\}, \tag{17}$$

which is a Bayesian approach that depends on the calculation of the posterior density of the parameters θ given the data Y according to

$$p(\theta \mid Y) = \frac{p(Y \mid \theta)p(\theta)}{p(Y)}, \tag{18}$$

where $p(\theta)$ is a prior that allows for incorporation of user knowledge (before observing the data) and $p(Y \mid \theta)$ is the usual data likelihood.

Not only does this estimate have an explicit formulation; it is also the minimum mean square error estimate in that for any other estimate $\hat{\beta} = f(Y)$ computed as any other measurable function f of the data Y, it holds that

$$\mathbf{E}\left\{\|\theta - \hat{\theta}\|^2\right\} \leq \mathbf{E}\left\{\|\theta - \hat{\beta}\|^2\right\}. \tag{19}$$

In this sense, the conditional mean (17) is the most accurate estimate. Furthermore, quantifications of estimation accuracy may be directly obtained via the marginal densities $p(\theta_i \mid Y)$ of individual parameter vector values θ_i.

Nevertheless, it is currently not widely used. There are no doubt philosophical reasons for this stemming from the well-known debate between frequentist and Bayesian perspectives on inference (Efron 2013).

Another key reason is that it is difficult to compute. It requires the evaluation of a multidimensional integral,

$$\mathbf{E}\{\theta \mid Y\} = \int \int \cdots \int \theta \, p(\theta \mid \mathcal{Y}_N) \, d\theta_1 \cdots d\theta_n \tag{20}$$

as does the computation of the marginal densities

$$p(\theta_i \mid \mathcal{Y}_N) = \int \cdots \int p(\theta \mid \mathcal{Y}_N) \, d\theta_1$$
$$\cdots d\theta_{i-1} d\theta_{i+1} \cdots d\theta_n. \tag{21}$$

Evaluating these quantities requires adding fundamentally new capability beyond efficient linear algebra support to system identification software. It involves adding capability for numerical integration.

Integration in one dimension is straightforward. The well-known and used Simpson's rule is remarkably efficient in that the relationship between the computational error and the number of grid points m obeys

$$\text{Error} = O(m^{-4}) \tag{22}$$

so that every order of magnitude increase in m delivers four extra digits of precision. However, (20) is an $n_\theta = \dim\{\theta\}$ dimensional integral, and m grid points on each of n_θ axes imply

$$M = m^{n_\theta} \tag{23}$$

function evaluations. This can blow up quite quickly, as illustrated in Fig. 2 for the case of only modest $m = 30$ grid points and with respect to the very simple problem of estimating a
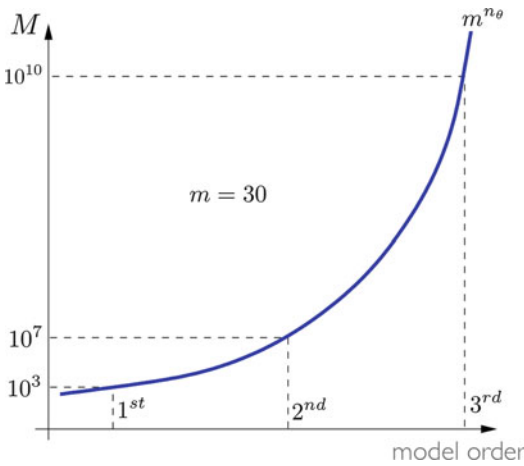
System Identification Software, Fig. 2 Increase in number of function evaluations M required for Simpson's rule integration with $m = 30$ grid points on each parameter axis associated with linear output-error models of increasing order. Note that accounting for both numerator and denominator parameters, $n_\theta = 2 \times$ model order $+ 1$

straightforward linear output-error model of increasing order.

On a serial CPU platform, there is an upper limit of time available to wait for a result and hence an upper limit M of function evaluations that are tolerable. Viewed as a function of this, the accuracy of simple Simpson's rule methods is

$$\text{Error} = O(M^{-4/n_\theta}), \qquad (24)$$

which is not attractive as model complexity and hence n_θ grows.

A further and vitally important problem is that it will generally not be clear where to allocate the m grid points on each axis since the support of the posterior $p(\theta \mid Y)$ is not readily known. Indeed, a main point of computing the multidimensional integrals associated with the marginals (21) is to determine this support.

A strategy to address these difficulties is based on the strong law of large numbers (SLLN). Namely, if random draws $x^i \sim p(x)$ from a density $p(x)$ can be obtained, then sample averages of functions of them converge with probability one to the ensemble average expectation, which is an integral:

$$\frac{1}{M} \sum_{i=1}^{M} f(x^i) \xrightarrow{\text{w.p.1}} \mathbf{E}\left\{f(x^i)\right\} = \int f(x)p(x)\,\mathrm{d}x. \qquad (25)$$

This principle may then be used as a "randomized" method to compute an estimate $\hat{I}_M$ of an integral I; viz.,

$$I = \int f(x)p(x)\,\mathrm{d}x \approx \hat{I}_M \triangleq \frac{1}{M}\sum_{i=1}^{M} f(x^i). \qquad (26)$$

Furthermore, if the x^i are independent draws, then

$$\text{Var}\{\hat{I}_M\} = \frac{1}{M^2}\sum_{i=1}^{M}\text{Var}\{f(x^i)\} = \frac{1}{M}\text{Var}\{f(x)\}, \qquad (27)$$

and hence the absolute error in integral evaluation is

$$O(|I - \hat{I}_M|) \approx O(M^{-1/2}). \qquad (28)$$

The vital point is that as opposed to (24), this error is *independent* of the dimension of x and hence *independent* of the dimension of the integral I. Furthermore, the grid points are the realizations $\{x^i\}$, which naturally will lie within the support of the integrand $f(x)p(x)$ and do not need to be otherwise designed.

Of course, this depends on a means to draw samples from an arbitrary density $p(\cdot)$ of interest, but simple methods such as the Metropolis–Hastings methods and "slice sampler" exist to achieve this Mackay (2003).

Importantly too, these randomized methods are ideally suited to exploiting the growing availability of desktop multicore computing platforms. Generating M realizations to form the integral approximation $\hat{I}_M$ in (26) may be achieved in one-tenth the time simply by running ten independently initialized random number generators in parallel, each generating one $M/10$ length realization. The method (26) is thus (in principal) trivial to parallelize.

Furthermore, much greater parallelization and hence also speedup may be achieved by employing the "graphics processing units" (GPUs) in desktop computers. These GPUs are inexpensive because they service a high volume

S

consumer demand for interactive gaming, which requires high-speed numerical computation for 3D-projected graphics. As such these GPUs have evolved to provide hundreds of parallel processing cores, each clocked in the gigahertz range.

To give an impression of the computational capability of GPU-based platforms, the single-precision giga-FLOPS (floating-point operations per second) performance history for NVIDIA brand GPUs and Intel architecture processors designed for desktop applications is profiled in Fig. 3.

This shows theoretical performance, assuming all cores may be fully utilized constantly. In reality, this is never possible due to communication and architecture restrictions. For example, GPU architectures are based on an SIMD (single instruction, multiple data) design, so at any one time many cores must execute the identical instruction, but may do so on different data. Analysis of these and other aspects relevant for system identification software implementation requires detailed study (Lee et al. 2010).

The fact that desktop hardware architectures have and will continue to offer more but not faster processing cores may be exploited in system identification software beyond this Bayesian setting. For example, the last decade has seen great interest in delivering estimation methods for an increasingly broad range of nonlinear model structures, a quite general version of which can be expressed in the nonlinear state–space form

$$x(t+1) \sim p(x(t+1) \mid x(t), \theta) \qquad (29)$$

$$y(t) \sim p(y(t) \mid x(t), \theta). \qquad (30)$$

In principle, there is no reason why this cannot be straightforwardly addressed by the usual maximum likelihood approach of forming the likelihood

$$p(Y_N \mid \theta) = \prod_{t=1}^{N} p(y(t) \mid Y_{t-1}, \theta),$$

$$Y_t = \{y(1), \cdots, y(t)\} \qquad (31)$$

and then using this as the cost function $V_N(\theta)$ in (1) and then proceeding with the usual gradient-based search. Indeed, there exist explicit formulae for computing the predictive densities $p(y(t) \mid Y_{t-1}, \theta)$ required in (31). Namely, the coupled measurement update

$$p(x(t) \mid Y_t, \theta) = \frac{p(y(t) \mid x(t), \theta)p(x(t) \mid Y_{t-1}, \theta)}{p(y(t) \mid Y_{t-1}, \theta)}$$

$$p(y(t) \mid Y_{t-1}, \theta) =$$
$$\int p(y(t) \mid x(t), \theta)p(x(t) \mid Y_{t-1}) \, \mathrm{d}x(t)$$

and time update

$$p(x(t+1) \mid Y_t, \theta) =$$
$$\int p(x(t+1) \mid x(t), \theta) \, p(x(t) \mid Y(t), \theta) \, \mathrm{d}x(t) \qquad (32)$$

equations.

However, again we are faced with the problem of numerically evaluating multidimensional integrals. The integral dimension this time is that of the state vector $x(t)$, which may be less than that of the parameter vector θ just discussed, but $2N$ of these integrals needs to be evaluated in order to compute the likelihood (31), and this needs to be redone for each step of any associated gradient-based search.

Again, a randomized algorithm approach based on the SLLN could be considered as a way forward in system identification software development. Indeed, sequential Monte Carlo (SMC) algorithms (aka particle filtering) (Doucet and Johansen 2011) have been specifically developed to compute the above integrals involved in the time and measurement update, and there has been recent work (Schön et al. 2011; Andrieu et al. 2010) on employing this to develop software for the estimation of the general nonlinear model (29) and (30).

The resulting algorithms are computationally intensive, to the point where implementation on serial CPU architectures means they are limited to deployment on nonlinear model structures of very low state dimension. However, again

System Identification Software, Fig. 3 Historical trend of theoretical single-precision giga- FLOPS performance of commodity NVIDIA brand GPUs versus Intel architecture CPUs designed for desktop computing
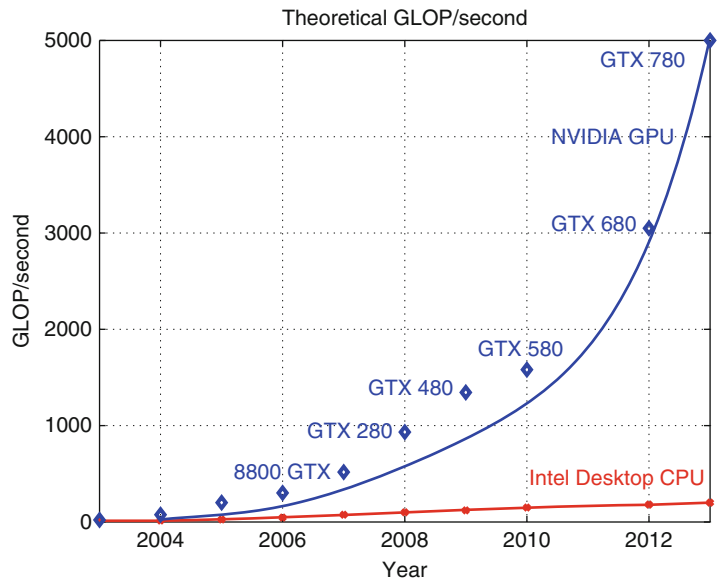
because the SLLN is at the heart of the methods, and averaging over one long run on a serial machine is numerically equivalent (but potentially much faster) to averaging over multiple shorter runs computed in parallel, there is scope for future system identification software to employ these approaches.

Examples of Available System Identification Software

With the features of current and perhaps future system identification software packages profiled, it may be useful to make specific mention of particular system identification software packages that have been under active development for a substantial period of time. These include the following commercially available packages:

1. The *MathWorks System Identification Toolbox* (Ljung 2012), which is arguably the most mature and comprehensive system identification software available
2. The *GAMAX Frequency Domain System Identification Toolbox* (Kollár et al. 2006), which specializes in estimation of models based on measurements in the frequency domain

3. The *Adaptx* software (Larimore 2000) specializing in the estimation of state–space models using subspace-based methods

Noncommercial and freely available system identification software packages that are relevant include:

1. The "computer-aided program for time-series analysis and identification of noisy systems" (CAPTAIN) toolbox (Young and Taylor 2012), which provides a platform supporting the "refined instrumental variable" (RIV) algorithm for linear system estimation;
2. The "continuous-time system identification" (CONTSID) toolbox (Garnier et al. 2012), which specializes in the estimation of continuous-time models
3. The "interactive software tool for system identification education" (ITSIE) toolbox (Guzmán et al. 2012), which has an emphasis on education and training in system identification principles
4. The "University of Newcastle identification toolbox" (UNIT) software (Ninness et al. 2013) that is designed as an open platform for researchers to evaluate the performance of new methods relative to established ones

Summary and Future Directions

A case can be mounted that at its heart, system identification is about the design of software and the understanding of the results provided by it. Certainly, the field has been built on decades of deep theoretical contributions, but this has been very practically focused either on delivering new algorithms that may be directly implemented or on better understanding the performance of existing algorithms.

Efficient numerical linear algebra routines have traditionally been the foundation of the resulting proven and effective system identification methods and software to date, and these have scaled in effectiveness as desktop computing clock speeds have scaled.

However, the recent past and the foreseeable future see CPU speed as static and with an increasing number of available processor cores. Delivering greater system identification capacity will require the development of methods whose software implementations can harness this growing availability of multiple processor cores.

Cross-References

▶ Frequency Domain System Identification
▶ Nonlinear System Identification Using Particle Filters
▶ System Identification: An Overview
▶ System Identification Techniques: Convexification, Regularization, and Relaxation

Recommended Reading

For readers wishing to gain a deeper understanding of the numerical linear algebra aspects discussed here, the classic text (Golub and Loan 1989) is recommended. Those wishing further background on the calculation of multidimensional integrals via randomized algorithms such as Metropolis–Hastings and slice sampling will find (Mackay 2003) useful. The particle filtering methods mentioned here for nonlinear estimation problems are clearly explained in Doucet

and Johansen (2011). Readers interested in further detail on numerical computations on GPU-based platforms supporting these computations will find (Lee et al. 2010) useful.

Bibliography

Andrieu C, Doucet A, Holenstein R (2010) Particle Markov chain Monte Carlo methods. J R Stat Soc Ser B 72:1–33

Doucet A, Johansen AM (2011) A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan D, Rozovsky B (eds) Nonlinear filtering handbook. Oxford University Press, London

Efron B (2013) A 250 year argument: belief, behaviour and the bootstrap. Bull Am Math Soc 50:129–146

Garnier H, Gilson M, Laurain V (2012) Developments for the CONTSID toolbox. In: Proceedings of the 16th IFAC symposium on system identification, Brussels. ISBN:978–3–902823–06–9

Golub G, Loan CV (1989) Matrix computations. Johns Hopkins University Press, Baltimore

Guzmán J, Rivera D, Dormido S, Berenguel M (2012) An interactive software tool for system identification. Adv Eng Softw 45:115–123

Hjalmarsson H, Sjöberg J (2012) A mathematica toolbox for signals, systems and identification. In: Proceedings of the 16th IFAC symposium on system identification, Brussels

Kollár I, Pintelon R, Schoukens J (2006) Frequency domain system identification toolbox for Matlab: characterizing nonlinear errors of linear models. In: Proceedings of the 14th IFAC symposium on system identification, Newcastle, pp 726–731

Larimore WE (2000) The adaptx software for automated multivariable system identification. In: Proceedings of the 12th IFAC symposium on system identification, Santa Barbara

Lee A, Yau C, Giles M, Doucet A, Holmes C (2010) On the utility of graphics cards to perform massively parallel simulation of advanced monte–carlo methods. J Comput Graph Stat 19: 769–789

Ljung L (1999) System identification: theory for the user, 2nd edn. Prentice-Hall, New Jersey

Ljung L (2012) MATLAB system identification toolbox users guide, version 8. The Mathworks

Mackay DJ (2003) Information theory, inference, and learning algorithms. Cambridge University Press, Cambridge/New York

Ninness B, Wills A, Mills A (2013) Unit: a freely available system identification toolbox. Control Eng Pract 21:631–644

Schön T, Wills A, Ninness B (2011) System identification of nonlinear state-space models. Automatica 37:39–49

Young PC, Taylor CJ (2012) Recent developments in the CAPTAIN toolbox for Matlab. In: Proceedings of the 16th IFAC symposium on system identification, Brussels. ISBN:978–3–902823–06–9

System Identification Techniques: Convexification, Regularization, and Relaxation

Alessandro Chiuso
Department of Information Engineering,
University of Padova, Padova, Italy

Abstract

System identification has been developed, by and large, following the classical parametric approach. In this entry we discuss how regularization theory can be employed to tackle the system identification problem from a nonparametric (or semi-parametric) point of view. Both regularization for smoothness and regularization for sparseness are discussed, as flexible means to face the bias/variance dilemma and to perform model selection. These techniques have also advantages from the computational point of view, leading sometimes to convex optimization problems.

Keywords

Kernel methods; Nonparametric methods; Optimization; Sparse Bayesian learning; Sparsity

Introduction

System identification is concerned with automatic model building from measured data. Under this unifying umbrella, this field spans a rather broad spectrum of topics, considering different model classes (linear, hybrid, nonlinear, continuous, and discrete time) as well as a variety of methodologies and algorithms, bringing together in a nontrivial way concepts from classical statistics, machine learning, and dynamical systems.

Even though considerable effort has been devoted to specific areas, such as parametric methods for linear system identification which are by now well developed (see the introductory article ▶ System Identification: An Overview), it is fair to say that modeling still is, by far, the most time-consuming and costly step in advanced process control applications. As such, the demand for fast and reliable automated procedures for system identification makes this exciting field still a very active and lively one.

Suffices here to recall that, following this classic parametric maximum likelihood (ML)/prediction error (PE) framework, the candidate models are described using a finite number of parameters $\theta \in \mathbb{R}^n$. After the model classes have been specified, the following two steps have to be undertaken:

(i) Estimate the model complexity $\hat{n}$.
(ii) Find the estimator $\hat{\theta} \in \mathbb{R}^{\hat{n}}$ minimizing a cost function $J(\theta)$, e.g., the prediction error or (minus) the log-likelihood.

Both of these steps are critical, yet for different reasons: step (ii) boils down to an optimization problem which, in general, is non-convex and as such it is very hard to guarantee that a global minimum is achieved. The regularization techniques discussed in this entry sometimes allow to reformulate the identification problem as a convex program, thus solving the issue of local minima.

In addition fixing the system complexity equal to the "true" one is a rather unrealistic assumption and in practice the complexity n has to be estimated as per step (i). In practice there is never a "true" model, certainly not in the model class considered. The problem of statistical modeling is first of all an approximation problem; one seeks for an approximate description of "reality" which is at the same time simple enough to be learned with the available data and also accurate enough for the purpose at hand. On this issue see also the section "Trade-off Between Bias and Variance" in ▶ System Identification: An Overview. This has nontrivial implications, chiefly the facts that classical order selection criteria are based on asymptotic arguments and that the statistical properties of estimators $\hat{\theta}$ after model selection, called post-model-selection estimators (PMSEs), are in general difficult to study (Leeb and Pötscher 2005) and may lead to undesirable behavior. Experimental evidence shows

S

that this is not only a theoretical problem but also a practical one (Pillonetto et al. 2011; Chen et al. 2012). On top of this statistical aspect, there is also a computational one. In fact the model selection step, which includes as special cases also variable selection and structure selection, may lead to computationally intractable combinatorial problems. Two simple examples which reveal the combinatorial explosion of candidate models are the following: (a) *Variable selection*: consider a high-dimensional time series (MIMO) where not all inputs/outputs are relevant and one would like to select k out of m available input signals where k is not known and needs to be inferred from data; (see, e.g., Banbura et al. (2010) and Chiuso and Pillonetto (2012)), and (b) *structure selection*: consider all autoregressive models of maximal lag p with only $p_0 < p$ nonzero coefficients and one would like to estimate how many (p_0) and which coefficients are nonzero. The same combinatorial problem arises in hybrid system identification (e.g., switching ARX models). Given that enumeration of all possible models is essentially impossible due the combinatorial explosion of candidates, selection could be performed using greedy approaches from multivariate statistics, such as stepwise methods (Hocking 1976).

The system identification community, inspired by work in statistics (Tibshirani 1996; Mackay 1994), machine learning (Rasmussen and Williams 2006; Tipping 2001; Bach et al. 2004), and signal processing (Donoho 2006; Wipf et al. 2011), has recently developed and adapted methods based on regularization to jointly perform model selection and estimation in a computationally efficient and statistically robust manner. Different regularization strategies have been employed which can be classified in two main classes: regularization induced by so-called smoothness priors (aka Tikhonov regularization; see Kitagawa and Gersh (1984) and Doan et al. (1984) for early references in the field of dynamical systems) and regularization for selection. This latter is usually achieved by convex relaxation of the ℓ_0 quasinorm (such as ℓ_1 norm and variations thereof such as sum of norms, nuclear norm, etc.) or other non-

convex sparsity-inducing penalties which can be conveniently derived in a Bayesian framework, aka sparse Bayesian learning (SBL) (Mackay 1994; Tipping 2001; Wipf et al. 2011).

The purpose of this entry is to guide the reader through the most interesting and promising results on this topic as well as areas of active research; of course this subjective view only reflects the author's opinion, and of course different authors could have offered a different perspective.

While, as mentioned above, system identification studies various classes of models (ranging from linear to general "nonlinear" models), in this entry, we shall restrict our attention to specific ones, namely, linear and hybrid dynamical systems. The field of nonlinear system identification is so vast (a quote sometimes attributed to S. Ulam has it that the study of nonlinear systems is a sort of "non-elephant zoology") that even though it has largely benefitted from the use of regularization, it cannot be addressed within the limited space of this contribution. The reader is referred to the Encyclopedia chapters ▶ Nonlinear System Identification: An Overview of Common Approaches and ▶ Nonlinear System Identification Using Particle Filters for more details on nonlinear model identification.

System Identification

Let $u_t \in \mathbb{R}^m$, $y_t \in \mathbb{R}^p$ be, respectively, the measured *input* and *output* signals in a dynamical system; the purpose of system identification is to find, from a finite collection of input-output data $\{u_t, y_t\}_{t \in [1,N]}$, a "good" dynamical model which describes the phenomenon under observation. The candidate model will be searched for within a so-called "model set" denoted by $\mathcal{M}$. This set can be described in parametric form (see, e.g., Eq. (3) in ▶ System Identification: An Overview) or in a nonparametric form. In this entry we shall use the symbol $\mathcal{M}_n(\theta)$ for parametric model classes where the subscript n denotes the model complexity, i.e., the number of free parameters.

Linear Models

The first part of the entry will address identification of linear models, i.e., models described by a convolution

$$y_t = \sum_{k=1}^{\infty} g_{t-k}u_k + \sum_{k=0}^{\infty} h_{t-k}e_k \quad t \in \mathbb{Z} \quad (1)$$

where g and h are the so-called impulse responses of the system and $\{e_t\}_{t\in\mathbb{Z}}$ is a zero-mean white noise process which under suitable assumptions is the one-step-ahead prediction error; a convenient description of the linear system (1) is given in terms of the transfer functions

$$G(q) := \sum_{k=1}^{\infty} g_k q^{-k} \quad H(q) := \sum_{k=0}^{\infty} h_k q^{-k}$$

The linear model (1) naturally yields an "optimal" (in the mean square sense) output predictor which shall be denoted later on by $\hat{y}_{t|t-1}$. As mentioned above, under suitable assumptions, the noise e_t in (1) is the so-called *innovation* process $e_t = y_t - \hat{y}_{t|t-1}$. See also Eq. (8) in ▶ System Identification: An Overview.

When g and h are described in a parametric form, we shall use the notation $g_k(\theta), h_k(\theta)$, and, likewise, $G(q,\theta), H(q,\theta)$, and $\hat{y}_{t|t-1}(\theta)$.

Example 7 Consider the so-called "output-error" model, i.e., assume $H(q) = 1$. An example of *parametric* model class is obtained restricting $G(q,\theta)$ to be a rational function

$$G(q,\theta) = K \prod_{i=1}^{n} \frac{q - z_i}{q - p_i}$$

where $\theta := [K, p_1, z_1, \ldots, p_n, z_n]$ is the parameter vector. Note that the parameter vector θ may subjected to constraints $\theta \in \Theta$, e.g., enforcing that the system be bounded input, bounded output (BIBO) stable ($|p_i| < 1$) or that the impulse response be real ($K \in \mathbb{R}$ and poles p_i and zeros z_i appear in complex conjugate pairs).

An example of *nonparametric* model is obtained, e.g., postulating that g_k is a realization of a Gaussian process (Rasmussen and Williams 2006) with zero mean and a certain covariance function $R(t,s) = \text{cov}(g_t, g_s)$. For instance, the choice $R(t,s) = \lambda^t \delta_{t-s}$, where $|\lambda| < 1$ and δ_k is the Kronecker symbol, postulates that the g_t and g_s are uncorrelated for $t \neq s$ and that the variance of g_t decays exponentially in t; this latter condition ensures that each realization g_k, $k > 0$, is BIBO stable with probability one. The exponential decay of g_t guarantees that, to any practical purpose, it can be considered zero for $t > T$ for a suitably large T. This allows to approximate the OE model with a "long" *finite impulse response* (FIR) model

$$G(q) = \sum_{k=1}^{T} g_k z^{-k} \quad (2)$$

where g_k, $k = 1, \ldots, T$, is modeled as a zero-mean Gaussian vector with covariance Σ, with elements $[\Sigma]_{ts} = R(t,s)$.

Remark 1 Note that the model (2), which has been obtained from truncation of a nonparametric model, could in principle be thought as a parametric model in which the parameter vector θ contains all the entries of g_k, $k = 1, \ldots, T$. Yet the truncation index T may have to be large even for relatively "simple" impulse responses; for instance, $\{g_k(\theta)\}_{k\in\mathbb{Z}^+}$ may be a simple decaying exponential, $g_k(\theta) = \alpha\rho^k$, which is described by two parameters (amplitude and decay rate), yet if $|\rho| \simeq 1$, the truncation index T needs to be large (ideally $T \to \infty$) to obtain sensible results (e.g., with low bias). Therefore, the number of parameters $T(m \times p)$ may be larger (and in fact much larger) than the available number of data points N. Under these conditions, the parameter θ cannot be estimated from any finite data segment unless further constraints are imposed.

The Role of Regularization in Linear System Identification

In order to simplify the presentation, we shall refer to the linear model (1) and assume that $H(q) = 1$, i.e., we consider the so-called linear output-error (OE) models. The extension to more

S

general model classes can be found in Pillonetto et al. (2011), Chen et al. (2012), Chiuso and Pillonetto (2012), and references therein.

The main purpose of regularization is to control the model complexity in a flexible manner, moving from families of rigid, finite dimensional parametric model classes $\mathcal{M}_n(\theta)$ to flexible, possibly infinite dimensional, models. To this purpose one starts with a "suitably large" model class which is constrained through the use of socalled regularization functionals. To simplify the presentation, we consider the FIR (2). The estimator $\hat{\theta}$ is found as the solution of the following optimization problem

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^n} J_F(\theta) + J_R(\theta; \lambda) \quad (3)$$

where $J_F(\theta)$ is the "fit" term often measured in terms of average squared prediction errors:

$$J_F(\theta) := \frac{1}{N} \sum_{t=1}^{N} \| y_t - \hat{y}_{t|t-1}(\theta) \|^2 \quad (4)$$

while $J_R(\theta; \lambda)$ is a regularization term which penalizes certain parameter vectors θ associated to "unlikely" systems. Equation (3) can be seen as a way to deal with the *bias-variance tradeoff*. The regularization term $J_R(\theta; \lambda)$ may depend upon some regularization parameters λ which need to be tuned using measured data. In its simplest instance,

$$J_R(\theta; \lambda) = \lambda J_R(\theta)$$

where λ is a scale factor that controls "how much" regularization is needed. We now discuss different forms of regularization $J_R(\theta; \lambda)$ which have been studied in the literature.

Example 8 Let us consider the FIR model in Eq. (2) and let θ be a vector containing all the unknown coefficients of the impulse response $\{g_k\}_{k=1,\dots,T}$. The linear least squares estimator

$$\hat{\theta}_{LS} := \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \| y_t - \hat{y}_{t|t-1}(\theta) \|^2 \quad (5)$$

is ill-posed unless the number of data N is larger (and in fact much larger) that the number of parameters T. From the statistical point of view, the estimator (5) would result for large T in small bias and large variance. The purpose of regularization is to render the inverse problem of finding θ from the data $\{y_t\}_{t=1,\dots,N}$ well posed, thus better trading bias versus variance. The simplest form of regularization is indeed the socalled ridge regression or its weighted version (aka generalized Tikhonov regularization), where the 2-norm of the vector θ is weighted w.r.t. a positive semidefinite matrix Q,

$$\hat{\theta}_{\text{Reg}} := \arg \min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \| y_t - \hat{y}_{t|t-1}(\theta) \|^2$$
$$+ \lambda \theta^\top Q \theta \quad (6)$$

which result in so-called regularization for smoothness; see section "Regularization for Smoothness." The choice of the weighting Q is highly nontrivial in the system identification context, and the performance of the regularized estimator $\hat{\theta}_{\text{Reg}}$ heavily depends on this.

Remark 2 In order to formalize these ideas for nonparametric models or, equivalently, when the parameter θ is infinite dimensional, one has to bring in functional analytic tools, such as reproducing kernel Hilbert spaces (RKHS). This is rather standard in the literature on ill-posed inverse problems and has been recently introduced also in the system identification setting (Pillonetto et al. 2011). We shall not discuss these issues here because, we believe, the formalism would render the content less accessible.

Note that this regularization approach admits a completely equivalent Bayesian formulation simply setting

$$p(y|\theta) \propto e^{-J_F(\theta)} \quad p(\theta|\lambda) \propto e^{-J_R(\theta; \lambda)} \quad (7)$$

The densities $p(y|\theta)$ and $p(\theta|\lambda)$ are, respectively, the likelihood function and the prior, which in turn may depend on the unknown regularization parameters λ, aka hyperparameters in this Bayesian formulation. This is straightforward in

the finite dimensional setting, while it requires some care when θ is infinite dimensional. With reference to Example 7, and assuming θ contains the impulse response coefficients g_k in (2), $p(\theta|\lambda)$ is a Gaussian density with zero mean and covariance Σ which may be depend upon some regularization parameters λ. From the definitions (7), it follows that

$$p(\theta|y, \lambda) \propto p(y|\theta) p(\theta|\lambda) \qquad (8)$$

from which point estimators of θ can be obtained (e.g., as posterior mean, MAP, etc.). As such, with some abuse of terminology, we shall indifferently refer to $J_R(\theta; \lambda)$ as the "regularization term" or the "prior." The unknown parameter λ is used to introduce some flexibility in the regularization term $J_R(\theta; \lambda)$ or equivalently in the prior $p(\theta|\lambda)$ and is tuned based on measured data as discussed later on.

The regularization term $J_R(\theta; \lambda)$ can be roughly classified in *regularization for smoothness*, which attempts to control complexity in a smooth fashion and *regularization for sparseness* which, on top of estimation, also aims at selecting among a finite (yet possibly very large) number of candidate model classes.

Regularization for Smoothness

Let us consider a single-input, single-output FIR model of length T (arbitrarily large) and let $\theta := [g_1 \ g_2 \ \dots g_T]^\top \in \mathbb{R}^T$ be the (finite) impulse response; define also $y \in \mathbb{R}^N$ be the vector of output observations, Φ the regressor matrix with past input samples, and e the vector with innovations (zero mean, variance $\sigma^2 I$). With this notation the convolution input-output equation (1) takes the form

$$y = \Phi\theta + e$$

Following the prescriptions of ridge regression, a regularized estimator $\hat{\theta}$ can be found setting

$$J_R(\theta; \lambda) = \theta^\top K^{-1}(\lambda)\theta \qquad (9)$$

where the matrix $K(\lambda)$, aka kernel, is tailored to capture specific properties of impulse responses (exponential decay, BIBO stability, smoothness, etc.). Early references include Doan et al. (1984) and Kitagawa and Gersh (1984), while more recent work can be found in Pillonetto and De Nicolao (2010), Pillonetto et al. (2011) and Chen et al. (2012) where several choices of kernels are discussed.

Example 9 The simplest example of kernel is the so-called "exponentially decaying" kernel

$$K(\lambda) := \gamma D(\rho) \quad D(\rho) := \text{diag}\{\rho, \dots, \rho^T\} \qquad (10)$$

where $\lambda := (\gamma, \rho)$ with $0 < \rho < 1$ and $\gamma \geq 0$.

For fixed λ, the estimator $\hat{\theta}(\lambda)$ is the solution of a quadratic problem and can be written in closed form (aka ridge regression):

$$\hat{\theta}(\lambda) = K(\lambda)\Phi^\top \left(\Phi K(\lambda)\Phi^\top + \sigma^2 I\right)^{-1} y \qquad (11)$$

Two common strategies adopted to estimate the parameters λ are cross validation (Ljung 1999) and marginal likelihood maximization. This latter approach is based on the Bayesian interpretation given in Eqs. (7) from which one can compute the so-called "empirical Bayes" estimator $\hat{\theta}_{\text{EB}} := \hat{\theta}(\hat{\lambda}_{\text{ML}})$ of θ plugging in (11) the estimator of λ which maximizes the marginal likelihood:

$$\hat{\lambda}_{\text{ML}} := \arg \max_\lambda p(\lambda|y)$$
$$= \arg \max_\lambda \int p(\lambda, \theta|y) \, d\theta \qquad (12)$$

The main strength of the marginal likelihood is that, by integrating the joint posterior over the unknown hyperparameters θ, it automatically accounts for the residual uncertainty in θ for fixed λ. When both J_F and J_R are quadratic costs, which corresponds to assuming that e and θ are independent and Gaussian, the marginal likelihood in (12) can be computed in closed form so that

$$\hat{\lambda}_{\mathrm{ML}} := \arg \min_{\lambda} \ \log(\det(\Sigma(\lambda)))$$

$$+ \ y^{\top} \Sigma^{-1}(\lambda) y$$

$$\Sigma(\lambda) := \Phi K(\lambda) \Phi^{\top} + \sigma^2 I \qquad (13)$$

It is here interesting to observe that $\hat{\lambda}_{\mathrm{ML}}$ which solves (12) under certain conditions leads to $K(\hat{\lambda}_{\mathrm{ML}}) = 0$ (see Example 10), so that the estimator of θ in (11) satisfies $\hat{\theta}(\hat{\lambda}_{\mathrm{ML}}) = 0$. This simple observation is the basis of so-called *sparse Bayesian learning* (SBL); we shall return to this issue in the next section when discussing regularization for sparsity and selection.

Unfortunately the optimization problem (12) (or (13)) is not convex and thus subjected to the issue of local minima. However, both experimental evidence and some theoretical results support the use of marginal likelihood maximization for estimating regularization parameters; see, e.g., Rasmussen and Williams (2006) and Aravkin et al. (2014).

Regularization for Sparsity: Variable Selection and Order Estimation

The main purpose of regularization for sparseness is to provide estimators $\hat{\theta}$ in which subsets or functions of the estimated parameters are equal to zero.

Consider the multi-input, multi-output OE model

$$y_{t,j} = \sum_{i=1}^{m} \sum_{k=1}^{T} g_{k,ij} u_{t-k,i} + e_{t,i} \quad j = 1, \dots, p \tag{14}$$

where $y_{t,j}$ denotes the jth component of $y_t \in \mathbb{R}^p$; let also $\theta \in \mathbb{R}^{T(m+p)}$ be the vector containing all the impulse response coefficients $g_{k,ij}$, $j = 1, \dots, p$, $i = 1, \dots, m$, and $k = 1, \dots, T$. With reference to Eq. (14), simple examples of sparsity one may be interested in are:

(i) Single elements of the parameter vector θ, which corresponds to eliminating specific lags of some variables from the model (14).

(ii) Groups of parameters such as the impulse response from ith input to the jth output

$g_{k,ij}$, $k = 1, \dots, T$, thereby eliminating the ith input from the model for the jth output.

(iii) The singular values of the Hankel matrix $\mathcal{H}(\theta)$ formed with the impulse response coefficients g_k; in fact the rank of the Hankel matrix equals the order (i.e., the McMillan degree) of the system. (Strictly speaking any full rank FIR model of length T has McMillan degree $T \times p$. Yet, we consider $\{g_k\}_{k=1,\dots,T}$ to be the truncation of some "true" impulse response $\{g_k\}_{k=1,\dots,\infty}$, and, as such, the finite Hankel matrix built with the coefficients g_k will have rank equal to the McMillan degree of $G(q) = \sum_{k=1}^{\infty} g_k z^{-k}$.)

To this purpose one would like to penalize the number of nonzero terms, let them be entries of θ, groups, singular values, etc. This is measured by the ℓ_0 quasinorm or its variations: group ℓ_0 and ℓ_0 quasinorm of the Hankel singular values, i.e., the rank of the Hankel matrix. Unfortunately if J_R is a function of the ℓ_0 quasinorm, the resulting optimization problem is computationally intractable; as such one usually resorts to relaxations. Three common ones are described below.

One possibility is to resort to greedy algorithms such as orthogonal matching pursuit; generically it is not possible to guarantee convergence to a global minimum point.

A very popular alternative is to replace the ℓ_0 quasinorm by its *convex envelope*, i.e., the ℓ_1 norm, leading to algorithms known in statistics as LASSO (Tibshirani 1996) or its group version Group LASSO (Yuan and Lin 2006):

$$J_R(\theta; \lambda) = \lambda \|\theta\|_1 \tag{15}$$

Similarly the convex relaxation of the rank (i.e., the ℓ_0 quasinorm of the singular values) is the so-called nuclear norm (aka Ky Fan n-norm or trace norm), which is the sum of the singular values $\|A\|_* := \mathrm{trace}\{\sqrt{A^{\top} A}\}$ where $\sqrt{\cdot}$ denotes the matrix square root which is well defined for positive semidefinite matrices. In order to control the order (McMillan degree) of a linear system, which is equal to the rank of the Hankel matrix $\mathcal{H}(\theta)$ built with the impulse response described

by the parameter θ, it is then possible to use the regularization term

$$J_R(\theta; \lambda) = \lambda \|\mathcal{H}(\theta)\|_* \qquad (16)$$

thus leading to convex optimization problems (Fazel et al. 2001). Both (16) and (15) induce sparse or nearly sparse solutions (in terms of elements or groups of θ (15) or in terms of Hankel singular values (16)), making them attractive for selection. It is interesting to observe that both ℓ_1 and group ℓ_1 are special cases of the nuclear norm if one considers matrices with fixed eigenspaces. Yet, as well documented in the statistics literature, both (16) and (15) do not provide a satisfactory trade-off between sparsity and shrinking, which is controlled by the regularization parameter λ. As λ varies one obtains the so-called *regularization path*. Increasing λ the solution gets sparser but, unfortunately, it suffers from shrinking of nonzero parameters. To overcome these problems, several variations of LASSO have been developed and studied, such as adaptive LASSO (Zou 2006), SCAD (Fan and Li 2001), and so on. We shall now discuss a Bayesian alternative which, to some extent, provides a better trade-off between sparsity and shrinking than the ℓ_1 norm.

This Bayesian procedure goes under the name of sparse Bayesian learning and can be seen as an extension of the Bayesian procedure for regularization described in the previous section. In order to illustrate the method, we consider its simplest instance. Consider an MIMO system as in (14) with $p = 1$ and $m = 2$, i.e.,

$$\begin{aligned} y_t &= \sum_{k=1}^{T} g_{k,1} u_{t-k,1} + \sum_{k=1}^{T} g_{k,2} u_{t-k,2} + e_t \\ &= \phi_{t,1}^\top g_1 + \phi_{t,2}^\top g_2 + e_t \end{aligned} \qquad (17)$$

where $g_i := [g_{1,i}, \ldots, g_{t,i}]$. Let $\theta := [g_1^\top \ g_2^\top]^\top$ and assume that the g_i's are independent Gaussian random vectors with zero mean and covariances $\lambda_i K$. Letting $\Phi_i := [\phi_{1,i}, \ldots, \phi_{N,i}]^\top$ and following the formulation in (7) and (8), it follows that the marginal likelihood estimator of λ takes the form

$$\hat{\lambda}_{\mathrm{ML}} := \arg \min_{\lambda_i \geq 0} \log(\det(\Sigma(\lambda))) + y^\top \Sigma^{-1}(\lambda) y$$

$$\Sigma(\lambda) := \lambda_1 \Phi_1 K \Phi_1^\top + \lambda_2 \Phi_2 K \Phi_2^\top + \sigma^2 I \qquad (18)$$

After $\hat{\lambda}_{\mathrm{ML}}$ has been found, the estimator of θ is found in closed form as per Eq. (11). It can be shown that under certain conditions on the observation vector y, the estimated hyperparameters $\hat{\lambda}_{\mathrm{ML},i}$ lie at the boundary, i.e., are exactly equal to zero. If $\hat{\lambda}_{\mathrm{ML},i} = 0$, then, from Eq. (11), also $\hat{g}_i = 0$; this reveals that in (17) the ith input does not enter into the model; see also Example 10 for a simple illustration.

These Bayesian methods for sparsity have been studied in a general regression framework in Wipf et al. (2011) under the name of "type-II" maximum likelihood. Further results can be found in Aravkin et al. (2014) which suggest that these Bayesian methods provide a better trade-off between sparsity and shrinking (i.e., are able to provide sparse solution without inducing excessive shrinkage on the nonzero parameters).

Remark 3 A more detailed analysis, see, for instance, Aravkin et al. (2014), shows that LASSO/GLASSO (i.e., ℓ_1 penalties) and SBL using the "empirical Bayes" approach can be derived under a common Bayesian framework starting from the joint posterior $p(\lambda, \theta|y)$. While SBL is derived from the maximization λ of the marginal posterior, LASSO/GLASSO corresponds to maximizing the joint posterior after a suitable change of variables. For reasons of space, we refer the interested reader to the literature for details.

Recent work on the use of sparseness for variable selection and model order estimation can be found in Wang et al. (2007), Chiuso and Pillonetto (2012); and references therein.

Example 10 In order to illustrate how sparse Bayesian learning leads to sparse solutions, we consider a very simplified scenario in which the measurements equation is

$$y_t = \theta u_{t-1} + e_t$$

where e_t is zero-mean, unit variance Gaussian and white and u_t is a deterministic signal. The purpose is to estimate the coefficient θ, which could be possibly equal to zero. Thus, the estimator should reveal whether u_{t-1} influences y_t or not.

Following the SBL framework, we model θ as a Gaussian random variable, with zero mean and variance λ, independent of e_t. Therefore, y_t is also Gaussian, zero mean, and variance $u_{t-1}^2\lambda+1$. Therefore, assuming N data points are available, the likelihood function for λ is given by

$$L(\lambda) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi(u_{t-1}^2\lambda + 1)}} e^{-\frac{1}{2}\sum_{i=1}^{N}\frac{y_t^2}{u_{t-1}^2\lambda + 1}}$$

Defining now

$$\hat{\lambda}_{\mathrm{ML}} := \arg\min_{\lambda \geq 0} -2\log L(\lambda)$$

one obtains that

$$\hat{\lambda}_{\mathrm{ML}} = \max(0, \lambda_*)$$

where λ_* is the solution of

$$\sum_{t=1}^{N} \frac{u_{t-1}^4\lambda + u_{t-1}^2\left(1 - y_t^2\right)}{u_{t-1}^2\lambda + 1} = 0$$

which unfortunately doesn't have a closed form solution. If however we assume that the input u_t is constant (without loss of generality say that $u_t = 1$), we obtain that

$$\lambda_* = \frac{1}{N}\sum_{t=1}^{N} y_t^2 - 1$$

thus

$$\hat{\lambda}_{\mathrm{ML}} = \max\left(0, \frac{1}{N}\sum_{t=1}^{N} y_t^2 - 1\right)$$

Clearly this is a threshold estimator which sets to zero $\hat{\lambda}_{\mathrm{ML}}$ when the sample variance of y_t

is smaller than the variance of e_t, which was assumed to be equal to 1. Thus, the empirical Bayes estimator of θ, as per Eq. (11), is given by

$$\hat{\theta} = \frac{\hat{\lambda}_{\mathrm{ML}}}{\sum_{i=1}^{N} u_{t-1}^2\hat{\lambda}_{\mathrm{ML}} + 1} \sum_{i=1}^{N} y_t u_{t-1}$$

which is clearly equal to zero when $\hat{\lambda}_{\mathrm{ML}} = 0$.

Extensions: Regularization for Hybrid Systems Identification and Model Segmentation

An interesting extension of linear systems is a class of so-called hybrid models described by a relation of the form

$$
\begin{aligned}
y_t &= \hat{y}_{\theta_k}(t|t-1) + e_t \\
\hat{y}_{\theta_k}(t|t-1) &= L_{\theta_k}(y_t^-, u_t^-) \\
\theta_k &\in \mathbb{R}^{n_k} \quad k = 1, \ldots, K
\end{aligned}
\tag{19}
$$

where the predictor $\hat{y}_{\theta_k}(t|t-1)$, which is a linear function $L_{\theta_k}(y_t^-, u_t^-)$ of the "past" histories $y_t^- := \{y_{t-1}, y_{t-2}, \ldots\}$ and $u_t^- := \{u_{t-1}, u_{t-2}, \ldots\}$, is parametrized by a parameter vector $\theta_k \in \mathbb{R}^{n_k}$; there are K different parameter vectors θ_k, $k = 1, \ldots, K$, whose evolution over time is determined by a so-called *switching mechanism*. The name *hybrid* hints at the fact that the model is described continuous-valued (y, u, and e) and discrete-valued (k) variables.

A well-studied subclass of (19) is composed by the so-called switching ARX models, where the predictor takes the special form

$$\hat{y}_{\theta_k}(t|t-1) = \phi_t^\top \theta_k \quad \theta_k \in \mathbb{R}^{n_k} \tag{20}$$

The regressor ϕ_t is a finite vector containing inputs u_s and outputs y_s in a finite past window $s \in [t-1, t-T]$, plus possibly a constant component to model changing "means." The value of $k \in [1, K]$ is determined by the switching mechanism $p(\phi_t, t) : \mathbb{R}^{n_k} \times \mathbb{R} \to \{1, \ldots, K\}$.

Two extreme but interesting cases are (i) $p(\phi_t, t) = p_t$, where $p(\cdot)$ is an exogenous and not measurable signal, and (ii) $p(\phi_t, t) = p(\phi_t)$,

where $p(\cdot)$ is an endogenous unknown measurable function of the regression vector ϕ_t. In any case, from the identification point of view, k at time t *is not* assumed to be known and, as such, the identification algorithm has to operate without knowledge of this switching mechanism.

Identification of systems in the form (20) requires to estimate (a) the number of models K and the position of the switches between different models, (b) the "dimension" of each model n_k, (c) the value of the parameters θ_k, and, possibly, (d) the function $p(\phi_t, t)$ which determines the switching mechanism.

Steps (b) and (c) are essentially as in section "System Identification" (see also the introductory paper ▸ System Identification: An Overview); however, this is complicated by steps (a) and (d), which in particular require that one is able to estimate, from data alone, which system is "active" at each time t.

Step (a), which is also related to the problem of *model segmentation*, has been tackled in the literature; see e.g., Ozay et al. (2012), Ohlsson and Ljung (2013), and references therein, by applying suitable penalties on the number of different models K and/or on the number of switches. Note that $p(\phi_t, t) \neq p(\phi_s, s)$ if and only if $\theta_t \neq \theta_s$. Based on this simple observation, one can construct a regularization which counts either the number of switches, i.e.,

$$J_R(\theta; \gamma) := \gamma \sum_{t=2}^{N} \|\|\theta_t - \theta_{t-1}\|\|_0, \quad (21)$$

or attempts to approximate the total number of different models computing

$$J_R(\theta; \gamma) := \gamma \sum_{t,s=1}^{N} w(s,t)\|\|\theta_t - \theta_s\|\|_0 \quad (22)$$

for a suitable weighting $w(t,s)$; see Ohlsson and Ljung (2013).

As discussed above, these quasinorms lead, in general, to unfeasible optimization problems (NP-hard). An exception is the case where one considers bounded noise, i.e., solves a problem of the form

$$\min_{\theta_t} \sum_{t=2}^{N} \|\theta_t - \theta_{t-1}\|_0 \quad s.t. \quad \|y_t - \phi_t^\top \theta_t\|_\infty < \epsilon \quad (23)$$

which is shown to be a convex problem; see Ozay et al. (2012). In general relaxations are used, typically using the ℓ_1/group-ℓ_1 penalties, thus relaxing (21) and (22) to

$$\begin{aligned} J_R(\theta; \lambda) &:= \lambda \sum_{t=2}^{N} \|\theta_t - \theta_{t-1}\|_1 \\ J_R(\theta; \lambda) &:= \lambda \sum_{t,s=1}^{N} w(s,t)\|\theta_t - \theta_s\|_1 \end{aligned} \quad (24)$$

This yields to the convex optimization problems:

$$\min_{\theta_t} \sum_t \left(y_t - \phi_t^\top \theta_t\right)^2 + \lambda \sum_{t=2}^{N} \|\theta_t - \theta_{t-1}\|_1 \quad (25)$$

or

$$\min_{\theta_t} \sum_t \left(y_t - \phi_t^\top \theta_t\right)^2 + \lambda \sum_{t,s=1}^{N} w(s,t)\|\theta_t - \theta_s\|_1 \quad (26)$$

Summary and Future Directions

We have presented a bird's eye overview of regularization methods in system identification. By necessity this overview was certainly incomplete and we encourage the reader to browse through the recent literature for new developments on this exciting topic; we hope the references we have provided are a good starting point. While regularization is quite an old topic, we believe it is fair to say that the nontrivial interaction between regularization and system theoretic concepts provides a wealth of interesting and challenging problems. Just to mention a few open questions: (i) how and why smoothness priors relate to system order (McMillan degree), (ii) how can one design kernels which, at the same time, are descriptive for dynamical systems and lead to computationally attractive problems suited for online identification, (iii) how should kernels for multi-output systems be designed, and (iv) which are the statistical properties of Bayesian

procedures such as SBL and its extensions in the context of system identification. Last but not least, while some results are available, nonlinear system identification still offers significant challenges.

Cross-References

▶ Nonlinear System Identification Using Particle Filters
▶ Nonlinear System Identification: An Overview of Common Approaches
▶ Subspace Techniques in System Identification
▶ System Identification: An Overview

Recommended Reading

The use of regularization methods for system identification can be traced back to the 1980s, see Doan et al. (1984) and Kitagawa and Gersh (1984); yet it is fair to say that the most significant developments are rather recent and therefore the literature is not established yet. The reader may consult Fazel et al. (2001), Pillonetto et al. (2011), Chen et al. (2012), Chiuso and Pillonetto (2012) and references therein. Clearly all this work has largely benefitted from cross fertilization with neighboring areas and, as such, very relevant work can be found in the fields of machine learning (Bach et al. 2004; Mackay 1994; Tipping 2001; Rasmussen and Williams 2006), statistics (Hocking 1976; Tibshirani 1996; Fan and Li 2001; Wang et al. 2007; Yuan and Lin 2006; Zou 2006), signal processing (Donoho 2006; Wipf et al. 2011) and econometrics (Banbura et al. 2010).

Bibliography

Aravkin A, Burke J, Chiuso A, Pillonetto G (2014) Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ARD and GLASSO. J Mach Learn Res 15:217–252

Bach F, Lanckriet G, Jordan M (2004) Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21st international conference on machine learning, Banff, pp 41–48

Banbura M, Giannone D, Reichlin L (2010) Large Bayesian VARs. J Appl Econom 25:71–92

Chen T, Ohlsson H, Ljung L (2012) On the estimation of transfer functions, regularizations and Gaussian processes – revisited. Automatica 48, pp 1525–1535

Chiuso A, Pillonetto G (2012) A Bayesian approach to sparse dynamic network identification. Automatica 48:1553–1565

Doan T, Litterman R, Sims C (1984) Forecasting and conditional projection using realistic prior distributions. Econom Rev 3:1–100

Donoho D (2006) Compressed sensing. IEEE Trans Inf Theory 52:1289–1306

Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96:1348–1360

Fazel M, Hindi H, Boyd S (2001) A rank minimization heuristic with application to minimum order system approximation. In: Proceedings of the 2001 American control conference, Arlington, vol 6, pp 4734–4739

Hocking RR (1976) A biometrics invited paper. The analysis and selection of variables in linear regression. Biometrics 32:1–49

Kitagawa G, Gersh H (1984) A smothness priors-state space modeling of time series with trends and seasonalities. J Am Stat Assoc 79:378–389

Leeb H, Pötscher B (2005) Model selection and inference: facts and fiction. Econom Theory 21:21–59

Ljung L (1999) System identification – theory for the user. Prentice Hall, Upper Saddle River

Mackay D (1994) Bayesian non-linear modelling for the prediction competition. ASHRAE Trans 100:3704–3716

Ohlsson H, Ljung L (2013) Identification of switched linear regression models using sum-of-norms regularization. Automatica 49:1045–1050

Ozay N, Sznaier M, Lagoa C, Camps O (2012) A sparsification approach to set membership identification of switched affine systems. IEEE Trans Autom Control 57:634–648

Pillonetto G, Chiuso A, De Nicolao G (2011) Prediction error identification of linear systems: a nonparametric Gaussian regression approach. Automatica 47:291–305

Pillonetto G, De Nicolao G (2010) A new kernel-based approach for linear system identification. Autonatica 46:81–93

Rasmussen C, Williams C (2006) Gaussian processes for machine learning. MIT, Cambridge

Tibshirani R (1996) Regression shrinkage and selection via the LASSO. J R Stat Soc Ser B 58:267–288

Tipping M (2001) Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244

Wang H, Li G, Tsai C (2007) Regression coefficient and autoregressive order shrinkage and selection via the LASSO. J R Stat Soc Ser B 69:63–78

Wipf D, Rao B, Nagarajan S (2011) Latent variable Bayesian models for promoting sparsity. IEEE Trans Inf Theory 57:6236–6255

Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R Stat Soc Ser B 68:49–67

Zou H (2006) The adaptive Lasso and it oracle properties. J Am Stat Assoc 101:1418–1429

System Identification: An Overview

Lennart Ljung
Division of Automatic Control, Department of Electrical Engineering, Linköping University, Linköping, Sweden

Abstract

This entry gives an overview of system identification. It outlines the basic concepts in the area and also serves as an umbrella contribution for the related nine articles on system identifications in this encyclopedia. The basis is the classical statistical approach of parametric methods using maximum likelihood and prediction error methods. The paper also describes the properties of the estimated models for large data sets.

Keywords

Asymptotic model properties; Dynamical systems; Estimation; Mathematica models; Maximum likelihood; Parameter estimates; Prediction error method; Regularization

An Introductory Example

System identification is the theory and art of estimating models of dynamical systems, based on observed inputs and outputs. Consider as a concrete example the Swedish aircraft fighter Gripen; see Fig. 1. From one of the earlier test flights, some data were recorded as depicted in Fig. 2.

To design the simulation software and the autopilot, the aircraft manufacturer, the SAAB company, needed a mathematical model for the dynamics of the system. It is a question to describe how, in this case, the pitch rate is affected by the three inputs. A fair amount of knowledge exists about aircraft dynamics, and in industrial practice, "gray-box" models based on Newton's laws of motion and unknown parameters like aerodynamical derivatives are employed to estimate the flight dynamics. Here, for the purpose of illustrating basic principles, let us just try a simple "black-box" difference equation relation. Denote the output, the pitch rate, at sample number t by $y(t)$, and three control inputs at the same time by $u_k(t), k = 1, 2, 3$. Then assume that we can write

$$
\begin{aligned}
y(t) = &- a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3) \\
&+ b_{1,1} u_1(t-1) + b_{1,2} u_1(t-2) \\
&+ b_{2,1} u_2(t-1) + b_{2,2} u_2(t-2) \\
&+ b_{3,1} u_3(t-1) + b_{3,2} u_3(t-2)
\end{aligned} \tag{1}
$$

In this simple relationship, we can adjust the parameters to fit the observed data as well as possible by a common least squares fit. We use only the 90 first data points of the observed data. That gives certain numerical values of the 9 parameters above:

$$
\begin{aligned}
&a_1 = -1.15, \, a_2 = 0.50, \, a_3 = -0.35, \\
&b_{1,1} = -0.54 \, b_{1,2} = 0.4, \, b_{2,1} = 0.15, \\
&b_{2,2} = 0.16, \, b_{3,1} = 0.16, \, b_{3,2} = 0.07
\end{aligned} \tag{2}
$$

We may note that this model is unstable – it has a pole at 1.0026, but that is in order, because the pitch channel of the real aircraft is unstable at the velocity and altitude in question.

How can we test if this model is reasonable? Since we used only half of the observed data for the estimation, we can test the model on the whole data record. Since the model is unstable it is natural to test it by letting it predict future outputs, say five samples ahead, and compare with the measured outputs. That is done in

S

**System Identification:
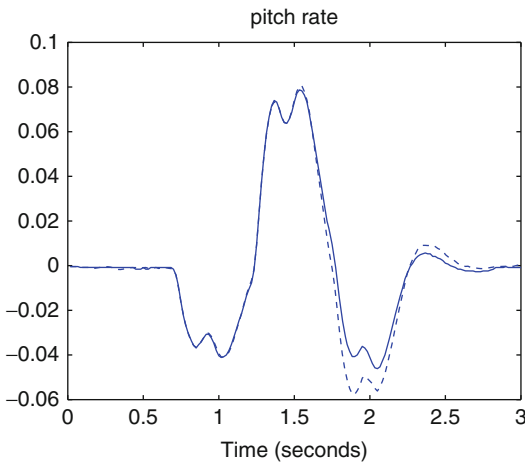An Overview, Fig. 1** The
Swedish aircraft Gripen
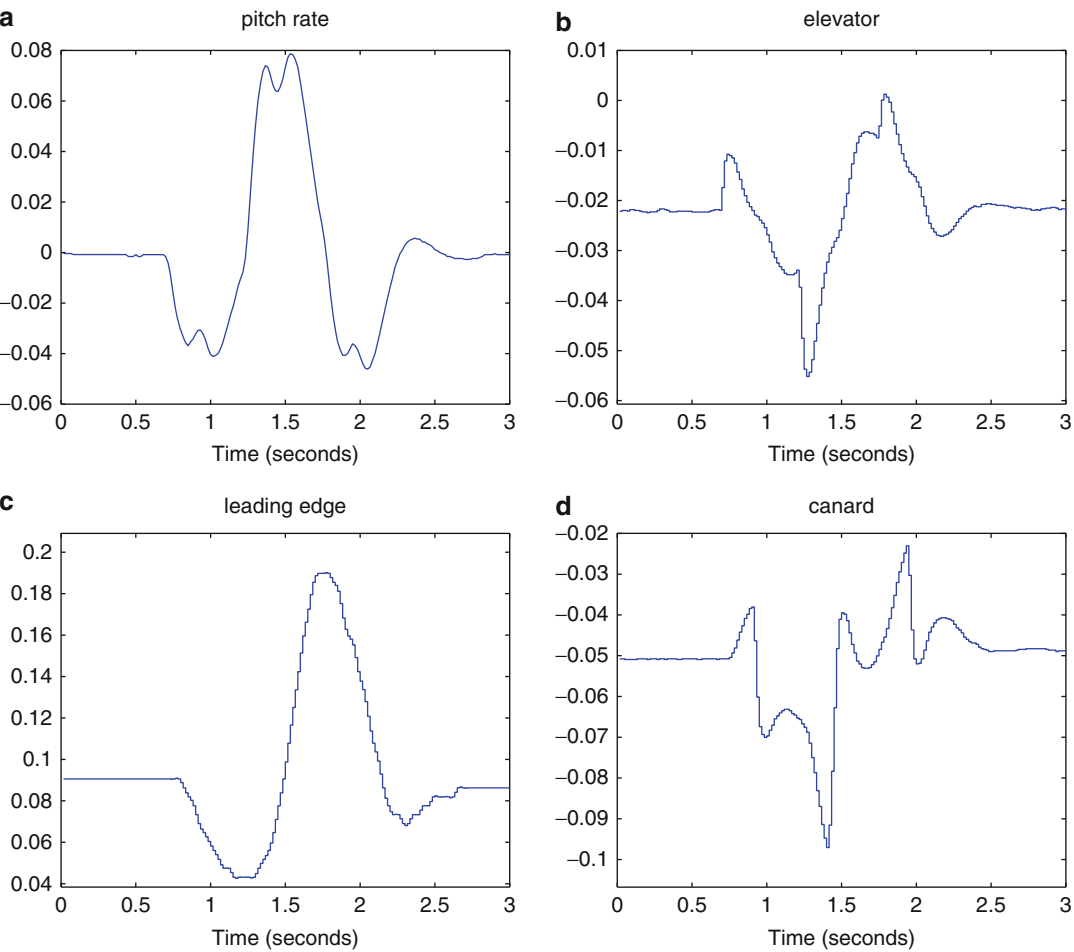
System Identification: An Overview, Fig. 2 Data from
an early test flight of Gripen. These data cover 3 s of
flight and are sampled at 60 Hz. (**a**) The output: pitch rate.
(**b**) Control input 1: elevator angle. (**c**) Control input 2:
leading edge flap. (**d**) Control input 3: canard angle

Fig. 3. We see that the simple model (2) provides
quite reasonable predictions over data it has
not seen before. This could conceivably be
improved if more elaborate model structures

than (1) were tried out. Also, in practice more
advanced techniques would be required to
validate that the estimated model is sufficiently
reliable.

System Identification: An Overview, Fig. 3 The measured output (*solid line*) compared to the 5-step-ahead prediction one (*dashed line*)

This simple introductory example points to the basic flow of system identification and it also points to the pertinent issues, which will be listed in the section "The State-of-the-Art Identification Setup."

Models and System Identification

The Omnipresent Model

It is clear to everyone in science and engineering that *mathematical models* are playing increasingly important roles. Today, model-based design and optimization is the dominant engineering paradigm to systematic design and maintenance of engineering systems. It has proven very successful and is widely used in basically all engineering disciplines. Concerning control applications, the aerospace industry is the earliest example on a grand-scale of this paradigm. This industry was very quick to adopt the theory for model-based optimal control that emerged in the 1960s and is spending great efforts and resources on developing models. In the process industry, model predictive control (MPC) has during the last 25 years become the dominant method to optimize production on an intermediate level. MPC uses dynamical models to predict future process behavior and to optimize the manipulated variables subject to process constraints.

Increasing demands on performance, efficiency, safety, and environmental aspects are pushing engineering systems to become increasingly complex. Advances in (wireless) communications systems and microelectronics are key enablers for this rapid development, allowing systems to be efficiently interconnected in networks, reducing costs and size, and paving the way for new sensors and actuators.

Model-based techniques are also gaining importance outside engineering applications. Let us just mention systems biology and health care. In the latter case it is expected that personalized health systems will become more and more important in the future.

Common to the examples given above are the requirements of permeating sensing, actuation, communication, and computation abilities of the engineering systems, in many cases in distributed architectures. It is also clear that these systems should be able to operate in a reliable way in an uncertain and temporally and spatially changing environment. In many applications, cognitive abilities and abilities to adapt will be important. With systems being decentralized and typically containing many actuators, sensors, states, and nonlinearities, but with limited access to sensor information, model building that delivers models of sufficient fidelity becomes very challenging.

System Identification: Data-Driven Modeling

Construction of models requires access to observed data. It could be that the model is developed entirely from information in signals from the system ("black-box models") or it could be that physical/engineering insights are combined with such information ("gray-box models"). In any case, verification (validation) of a model must be done in the light of measured data. Theories and methodologies for such model construction have been developed in many different research communities (to some extent independently). *System identification* is the term used in the control community for

the area of constructing mathematical models of dynamical systems from measured input-output signals. Other communities use other terms for often very similar techniques. The term *machine learning* has become very common in recent years, e.g., Rasmussen and Williams (2006).

System identification has a history of more than 50 years, since the term was coined by Lotfi Zadeh (1956). It is a mature research field with numerous publications, textbooks, conference series, and software packages. It is often used as an example in the control field of an area with good interaction between theory and industrial practice. The backbone of the theory relies upon statistical grounds, with maximum likelihood methods and asymptotic analysis (in the number of observed data). The goal of the system identification field is to find a model of the plant in question as well as of its disturbances and also to find a characterization of the uncertainty bounds of the description.

The State-of-the-Art Identification Setup

To approach a system identification problem, like in section "An Introductory Example," a number of questions need to be answered, such as
- What model type, e.g., (1) should be used?
- How should the parameters in the model be adjusted?
- What inputs should be applied to obtain a good model?
- How do we assess the quality of the model?
- How do we gain confidence in an estimated model?

There is a very extensive literature on the subject, with many textbooks, like Ljung (1999), Söderström and Stoica (1989), and Pintelon and Schoukens (2012).

System identification is characterized by five basic concepts:
- $\mathcal{X}$: The experimental conditions under which the data is generated
- $\mathcal{D}$: The data
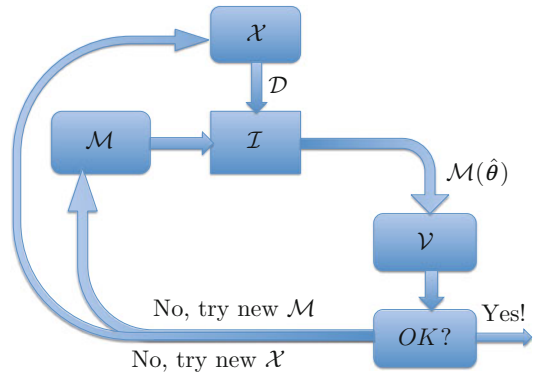
System Identification: An Overview, Fig. 4 The identification work loop

- $\mathcal{M}$: The model structure and its parameters θ
- $\mathcal{I}$: The identification method by which a parameter value $\hat{\theta}$ in the model structure $\mathcal{M}(\theta)$ is determined based on the data $\mathcal{D}$
- $\mathcal{V}$: The validation process that scrutinizes the identified model

See Fig. 4. It is typically an iterative process to navigate to a model that passes through the validation test ("is not falsified"), involving revisions of the necessary choices. For several of the steps in this loop, helpful support tools have been developed. It is however not quite possible or desirable to fully automate the choices, since subjective perspectives related to the intended use of the model are very important.

$\mathcal{M}$: Model Structures

A model structure $\mathcal{M}$ is a parameterized collection of models that describe the relations between the inputs u and outputs y of the system. The parameters are denoted by θ so $\mathcal{M}(\theta)$ is a particular model. The model set then is

$$\mathcal{M} = \{\mathcal{M}(\theta) | \theta \in D_{\mathcal{M}}\} \qquad (3)$$

Many ways exist to collect mathematical expressions that encompass a model; see, e.g., ▶ Modeling of Dynamic Systems from First Principles, ▶ Nonlinear System Identification: An Overview of Common Approaches, and

▶ Nonlinear System Identification Using Particle Filters. The models may be both linear and nonlinear as well as time invariant and time varying, and it is useful to have as a common ground that a model gives a rule to predict (one-step-ahead) the output at time t, i.e., $y(t)$ (a p-dimensional column vector), based on observations of previous input-output data up to time $t - 1$ (denoted by Z^{t-1}).

$$\hat{y}(t|\theta) = g(t, \theta, Z^{t-1}) \qquad (4)$$

This covers a broad variety of model descriptions, sometimes in a somewhat abstract way. The descriptions become much more explicit when we specialize to linear models.

A note on "inputs" It is important to include all measurable disturbances that affect y among the inputs u to the system, even if they cannot be manipulated as control inputs. In some cases the system may entirely lack measurable inputs, so the model (4) then just describes how future outputs can be predicted from past ones. Such models are called *time series* and correspond to systems that are driven by unobservable disturbances. Most of the techniques described in this entry apply also to such models.

A note on disturbances A complete model involves both a description of the input-output relations and a description of how various noise sources affect the measurements. The noise description is essential to understand both the quality of the model predictions and the model uncertainty. Proper control design also requires a picture of the disturbances in the system.

Linear Models

For linear time invariant systems, a general model structure is given by the transfer function G from input u to output y and the transfer function H from a white noise source e to output additive disturbances (for notational convenience, we specialize to single-input-single-output systems, but all expressions are valid in the multivariable case with simple notational changes):

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t) \qquad (5a)$$

$$Ee^2(t) = \sigma^2; \quad Ee(t)e^T(k) = 0 \text{ if } k \neq t \qquad (5b)$$

(E denotes mathematical expectation.) This model is in discrete time and q denotes the shift operator $qy(t) = y(t + 1)$. We assume for simplicity that the sampling interval is a one-time unit. The expansion of $G(q, \theta)$ in the inverse (backwards) shift operator gives the *impulse response* of the system:

$$G(q, \theta)u(t) = \sum_{k=1}^{\infty} g_k(\theta)q^{-k}u(t)$$

$$= \sum_{k=1}^{\infty} g_k(\theta)u(t - k) \qquad (6)$$

The discrete time Fourier transform (or the z-transform of the impulse response, evaluated in $z = e^{i\omega}$) gives the *frequency response* of the system:

$$G(e^{i\omega}, \theta) = \sum_{k=1}^{\infty} g_k(\theta)e^{-ik\omega} \qquad (7)$$

The function G describes how an input sinusoid shifts phase and amplitude when it passes through the system.

The additive noise term $v = He$ is quite versatile, and with a suitable choice of H, it can describe a disturbance with arbitrary spectrum. To link with the predictor as a unifying model concept, it is useful to compute the predictor for (5a) (the conditional mean of $y(t)$ given past data), which is

$$\hat{y}(t|\theta) = G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]$$
$$[y(t) - G(q, \theta)u(t)] \qquad (8)$$

Note that the expansion of H^{-1} starts with "1," so the first term starts with $h_1 q^{-1}$ so there is a delay in y. It is easy to interpret the first term as a simulation using the input u, adjusted with a prediction of the additive disturbance $v(t)$ at

time t, based on past values of v. The predictor is thus an easy reformulation of the basic transfer functions G and H. The question now is how to parameterize these.

Black-Box Models

A *black-box* model uses no physical insight or interpretation, but is just a general and flexible parameterization. It is natural to let G and H be rational in the shift operator:

$$G(q, \theta) = \frac{B(q)}{F(q)}; \quad H(q, \theta) = \frac{C(q)}{D(q)} \quad \text{(9a)}$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \ldots b_{nb} q^{-nb} \quad \text{(9b)}$$

$$F(q) = 1 + f_1 q^{-1} + \ldots + f_{nf} q^{-nf} \quad \text{(9c)}$$

$$\theta = [b_1, b_2, \ldots, f_{nf}] \quad \text{(9d)}$$

C and D are like F *monic*, i.e., start with a "1."

A very common case is that $F = D = A$ and $C = 1$ which gives the *ARX model* (autoregressive with exogenous input):

$$y(t) = A^{-1}(q)B(q)u(t) + A^{-1}(q)e(t) \text{ or} \quad \text{(10a)}$$

$$A(q)y(t) = B(q)u(t) + e(t) \text{ or} \quad \text{(10b)}$$

$$y(t) + a_1 y(t-1) + \ldots + a_{na} y(t - na) \quad \text{(10c)}$$

$$= b_1 u(t-1) + \ldots + b_{nb} u(t - nb) \quad \text{(10d)}$$

This is the model structure we used in (1) in the introductory example, but for several inputs.

Other common black-box structures of this kind are FIR (finite impulse response model, $F = C = D = 1$), ARMAX (autoregressive moving average with exogenous input, $F = D = A$), and BJ (Box-Jenkins, all four polynomials are different.)

Gray-Box Models

If some physical facts are known about the system, it is possible to build that into a *gray-box model*. It could, for example, be that for the airplane in the introduction, the motion equations are known from Newton's laws, but certain parameters are unknown, like the aerodynamical derivatives. Then it is natural to build a continuous-time state-space model from physical equations:

$$\dot{x}(t) = A(\theta)x(t) + B(\theta)u(t)$$
$$y(t) = C(\theta)x(t) + D(\theta)u(t) + v(t) \quad \text{(11)}$$

Here θ are simply some entries of the matrices A, B, C, D, corresponding to unknown physical parameters, while the other matrix entries signify known physical behavior. This model can be sampled with well-known sampling formulas (obeying the input inter-sample properties, zero-order hold, or first-order hold) to give

$$x(t + 1) = \mathcal{F}(\theta)x(t) + \mathcal{G}(\theta)u(t)$$
$$y(t) = C(\theta)x(t) + D(\theta)u(t) + w(t) \quad \text{(12)}$$

The model (12) has the transfer function from u to y

$$G(q, \theta) = C(\theta)[qI - \mathcal{F}(\theta)]^{-1}\mathcal{G}(\theta) + D(\theta) \quad \text{(13)}$$

so we have achieved a particular parameterization of the general linear model (5a).

Continuous-Time Models

The general model description (4) describes how the predictions evolve in discrete time. But in many cases, we are interested in continuous-time (CT) models, like models for physical interpretation and simulation (e.g., electrical circuit simulators like ADS, Spice, Spectre, and Microwave Office use continuous-time models). But CT model estimation is contained in the described framework, as the linear state-space model (11) illustrates. More comments on direct estimation of CT models are given in section "Estimating Continuous Time Models."

Nonlinear Models

A nonlinear model is a relation (4), where the function g is nonlinear in the input-output data Z. There is a rich variation in how to specify the

function g more explicitly. A quite general way is the nonlinear state-space equation, which is a counterpart to (12):

$$x(t + 1) = f(x(t), v(t), \theta)$$
$$y(t) = h(x(t), e(t), \theta)$$
(14)

where v and e are white noises. This is further discussed in ▸ Nonlinear System Identification: An Overview of Common Approaches, where x is described as a Markov process with v defining the transitions, and in ▸ Nonlinear System Identification: An Overview of Common Approaches, where (14) ($v \equiv 0$) is related to a continuous-time gray-box model. The latter article also discusses several other nonlinear model structures that can be seen as extensions and modifications of linear models: nonlinear mappings of past input-output data corresponding to (10), mixing static nonlinearities with linear dynamical models, etc.

$\mathcal{I}$: Identification Methods: Criteria

The goal of identification is to match the model to the data. Here the basic techniques for such matching will be discussed.

Time Domain Data

Suppose now we have collected a data record in the time domain

$$Z^N = \{u(1), y(1), \ldots, u(N), y(N)\}$$
(15)

Since the model is in essence a predictor, it is quite natural to evaluate it by how well it predicts the measured output. So, form the prediction errors for (4):

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta)$$
(16)

The "size" of this error can be measured by some scalar norm:

$$\ell(\varepsilon(t, \theta))$$
(17)

and the performance of the predictor over the whole data record Z^N becomes

$$V_N(\theta) = \sum_{t=1}^{N} \ell(\varepsilon(t, \theta))$$
(18)

A natural parameter estimate is then

$$\hat{\theta}_N = \arg\min_{\theta \in D_{\mathcal{M}}} V_N(\theta)$$
(19)

This is the *prediction error method (PEM)* and is applicable to general model structures. See, e.g., Ljung (1999) or (2002) for more details. See also ▸ Nonlinear System Identification: An Overview of Common Approaches.

The PEM approach can be embedded in a statistical setting to guarantee optimal statistical properties. The ML methodology below offers a systematic framework to do so:

A Maximum Likelihood View

If the system innovations e have a probability density function (pdf) $f(x)$, then the criterion function (18) with $\ell(x) = -\log f(x)$ will be the logarithm of the *likelihood function*. See Lemma 5.1 in Ljung (1999). More specifically, assume that the system has p outputs and that the innovations are Gaussian with zero mean and covariance matrix Λ, so that

$$y(t) = \hat{y}(t|\theta) + e(t), \quad e(t) \in N(0, \Lambda) \quad (20)$$

for the θ that generated the data. Then it follows that the negative logarithm of the likelihood function for estimating θ from y is

$$L_N(\theta) = \frac{1}{2}[V_N(\theta) + N \log \det \Lambda + Np \log 2\pi]$$
(21)

where $V_N(\theta)$ is defined by (18), with

$$\ell(\varepsilon(t, \theta)) = \varepsilon^T(t, \theta)\Lambda^{-1}\varepsilon(t, \theta)$$
(22)

So the maximum likelihood model estimate (MLE) for known Λ is obtained by minimizing $V_N(\theta)$. If Λ is not known, it can be included

S

among the parameters and estimated, (Ljung 1999, page 218), which results in a criterion

$$D_N(\theta) = \det \sum_{t=1}^{N} \varepsilon(t, \theta)\varepsilon^T(t, \theta) \qquad (23)$$

to be minimized.

The EM Algorithm

The EM algorithm (Dempster et al. 1977) is closely related to the ML technique. It is a method that is especially useful when the ML criterion is difficult to evaluate from the observed data but would be easier to find if certain unobserved *latent* variables were known. The algorithm alternates between an expectation step estimating the log likelihood and a maximization step bringing the parameter estimate closer in each step to the MLE. Its application to the nonlinear state-space model (14) is described in ▶ Nonlinear System Identification: An Overview of Common Approaches.

Regularization

Solving for the estimate in (19) is a so-called *inverse problem*, which means that the solution may be ill conditioned. To deal with that in (18), we could add a quadratic norm:

$$W_N(\theta) = V_N(\theta) + \lambda(\theta - \theta^\dagger)^T R(\theta - \theta^\dagger) \quad (24)$$

(λ is a scaling, R is a positive semidefinite (psd) matrix, and $\theta^\dagger$ is a nominal value of the parameters). The estimate is then found by minimizing $W_N(\theta)$. The criterion (24) makes sense in a classical estimation framework as an ad hoc modification of the MLE to deal with possible ill-conditioned minimization problems. The added quadratic term then serves as proper *(Tikhonov) regularization* of an ill-conditioned inverse problem; see, for example, Tikhonov and Arsenin (1977). This criterion is a clear-cut balance between model fit and a penalty on the model parameter size. The amount of penalty is governed by λ and R.

Other useful regularization penalties could be to add an ℓ_1 norm of the parameter. Such techniques are further discussed in ▶ System Identification Techniques: Convexification, Regularization, and Relaxation.

Bayesian View

For a broader perspective it is useful to invoke a Bayesian view. Then the sought parameter vector θ is itself a random vector with a certain pdf. This random vector will of course be correlated with the observations y. If we assume that the *prior distribution* of θ (before y has been observed) is Gaussian with mean θ^* and covariance matrix Π,

$$\theta \in N(\theta^*, \Pi) \qquad (25)$$

its prior pdf is

$$P(\theta) = \frac{1}{\sqrt{(2\pi)^p \det(\Pi)}} e^{-(\theta - \theta^*)^T \Pi^{-1}(\theta - \theta^*)/2} \qquad (26)$$

The posterior (after y has been measured) pdf then is by Bayes rule (Y denoting all measured y signals)

$$P(\theta|Y) = \frac{P(\theta, Y)}{P(Y)} = \frac{P(Y|\theta)P(\theta)}{P(Y)} \qquad (27)$$

In the last step $P(Y|\theta)$ is the likelihood function (cf. the negative log likelihood function $L_N(\theta)$ in (21)), $P(\theta)$ is the prior pdf (26), and $P(Y)$ is a θ-independent normalization. Apart from this normalization, and other θ-independent terms, twice the negative logarithm of (27) equals $W_N(\theta)$ in (24) with

$$\lambda R = \Pi^{-1} \qquad (28)$$

That means that with (28), the regularized estimate from (24) is the *maximum a posteriori* (MAP) estimate. As more and more data become available, the role of the prior will tend to zero, so as $N \to \infty$ the MAP Estimate $\to$ MLE.

This Bayesian interpretation of the regularized estimate also gives a clue to select the regularization quantities λ, R, θ^*.

For black-box models, a reasonable prior (Π, θ^*) may not be available. Then it is possible to parameterize them with *hyperparameters* α and then estimate these through the marginal likelihood:

$$\hat{\alpha} = \arg \max P(Y|\alpha) \qquad (29)$$

A survey of how such techniques may improve system identification techniques is given in Pillonetti et al. (2014).

More aspects of the Bayesian view of system identification are given in ▸ System Identification Techniques: Convexification, Regularization, and Relaxation and in ▸ Nonlinear System Identification Using Particle Filters.

Frequency Domain Data

Frequency domain data are obtained either from frequency analysers or by applying the Fourier transform to measured time domain data. The data could be in the input-output form

$$Y_N(e^{i\omega_k}), U_N(e^{i\omega_k}), \ k = 1, 2, \ldots, M \qquad (30)$$

$$Y_N(z) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} y(k) z^{-k} \qquad (31)$$

or being observed samples from the frequency function

$$\hat{\hat{G}}_N(e^{i\omega_k}), \ k = 1, 2, \ldots, M \qquad (32)$$

$$\text{e.g., } \hat{\hat{G}}_N(e^{i\omega}) = \frac{Y_N(e^{i\omega})}{U_N(e^{i\omega})} \text{ (ETFE)} \qquad (33)$$

((33) is the *empirical transfer function estimate*, ETFE).

Linear Parametric Models

By taking the Fourier transform of (5a), we see that

$$Y(e^{i\omega}) = G(e^{i\omega}, \theta) U(e^{i\omega}) \qquad (34)$$

plus a noise term that has variance

$$\sigma^2 |H(e^{i\omega}, \theta)|^2 \qquad (35)$$

Simple least squares (LS) curve fitting of (34) says that we should fit observations with weights that are inversely proportional to the measurement variance. That gives the weighted LS criterion

$$V_N(\theta) = \sum_{k=1}^{M} |Y(e^{i\omega_k}) \\ - G(e^{i\omega_k}, \theta) U_N(e^{i\omega_k})|^2 / |H(e^{i\omega_k}, \theta)|^2 \qquad (36)$$

(the constant σ^2 does not affect the minimization of V_N).

It can readily be verified that (36) coincides with (18), $(\ell(\varepsilon) = |\varepsilon|^2)$ by Parseval's identity in case $M = N$ and the frequencies ω_k are selected as the DFT grid.

Notice that (36) can be written as

$$V_N(\theta) = \sum_{k=1}^{M} \left| \frac{Y_N(e^{i\omega_k})}{U_N(e^{i\omega_k})} - G(e^{i\omega_k}, \theta) \right|^2 \\ \cdot \left| \frac{U_N(e^{i\omega_k})}{H(e^{i\omega_k}, \theta)} \right|^2 \qquad (37)$$

We can see that as a properly weighted curve fitting of the frequency function to the ETFE (33).

See ▸ Frequency Domain System Identification for more details of using frequency domain data for estimating dynamical systems.

Nonparametric Methods

From frequency domain data, the frequency response functions $G(e^{i\omega}), H(e^{i\omega})$ can also be estimated directly as functions without any parametric model. See ▸ Nonparametric Techniques in System Identification for a detailed account of this.

IV and Subspace Methods

Instrumental Variables

The family of identification methods that can be described as minimizing a specific criterion function, like (19), covers many theoretically and practically important techniques. Still, several methods do not belong to this family. A useful

S

technique is to characterize a good model, as one that gives prediction errors that are uncorrelated with available information:

$$\hat{\theta} = \mathrm{sol}_{\theta \in D_{\mathcal{M}}} \sum_{t=1}^{N} \varepsilon(t, \theta)\zeta(t, \theta) = 0 \qquad (38)$$

Here, $\varepsilon(t, \theta)$ is the prediction error (16), and sol means "solution to." The sequence $\{\zeta(t), t = 1, \ldots, N\}$ is constructed from the observed data, possibly also dependent on some design variables that are included in θ. Typically $\zeta(t)$ is constructed from past inputs, so a good model should not have prediction errors that are correlated with past observations. The variables ζ are called *instrumental variables*, and there is an extensive literature about how to select these. See, e.g., Ljung (1999), Section 7.5, Söderström and Stoica (1983), and Young (2011).

Subspace Methods

A related technique is to estimate black-box state-space models like (12) (without any internal parametric structure) by realizing the states from data and then estimating the matrices by least squares method. This gives a powerful family of methods for state-space model estimation. They are described in detail in ▶ Subspace Techniques in System Identification. The major advantage of subspace methods is that they easily apply to multiple-input-multiple-output systems and are non-iterative. A drawback is that the model properties and their dependence on certain design variables are not fully known.

Errors-in-Variables (EIV) Techniques

The estimation techniques described so far assume that the input has been measured without errors. In some cases, it is natural to assume that both inputs and outputs have measurement errors. The estimation problem then becomes more difficult, and some kind of knowledge about the measurement errors is typically required. In Pintelon and Schoukens (2012), Section 8.2, it is described how criteria of the type (36) are modified in the presence of input noise, and Söderström (2007) can be consulted for a summarizing treatise on EIV techniques. See also the section "Errors-in-Variables Framework" in ▶ Frequency Domain System Identification.

Asymptotic Properties of the Estimated Models

An estimated model is useless, unless something is known about its reliability and error bounds. Therefore, it is important to analyze the model properties.

Bias and Variance

The observations, certainly of the output from the system, are affected by noise and disturbances, which of course also will influence the estimated model parameters (19). The disturbances are typically described as stochastic processes, which makes the estimate $\hat{\theta}_N$ a *random variable*. This has a certain pdf and often the analysis is restricted to its mean and variance only. The difference between the mean and a true description of the system measures the *bias* of the model. If the mean coincides with the true system, the estimate is said to be *unbiased*. The total error in a model thus has two contributions: the bias and the variance.

Properties of the PEM Estimate (19) as $N \to \infty$

Except in simple special cases, it is quite difficult to compute the pdf of the estimate $\hat{\theta}_N$. However, its *asymptotic properties* as $N \to \infty$ are easier to establish. The basic results can be summarized as follows (E denotes mathematical expectation; see Ljung (1999), chapters 8 and 9, for a more complete treatment):

- **Limit Model:**

$$\hat{\theta}_N \to \theta^*$$

$$= \arg\min \left[\lim_{N \to \infty} \frac{1}{N} V_N(\theta) \approx \mathrm{E}\ell(\varepsilon(t, \theta)) \right] \tag{39}$$

So the estimate will converge to the best possible model, in the sense that it gives the smallest average prediction error.

- **Asymptotic Covariance Matrix for Scalar Output Models:**
  In case the prediction errors $e(t) = \varepsilon(t, \theta^*)$ for the limit model are approximately white, the covariance matrix of the parameters is asymptotically given by:

$$\text{Cov}\hat{\theta}_N \sim \frac{\kappa(\ell)}{N} \left[\text{Cov} \frac{d}{d\theta} \hat{y}(t|\theta) \right]^{-1} \quad (40)$$

So the covariance matrix of the parameter estimate is given by the inverse covariance matrix of the gradient of the predictor wrt the parameters. Here (prime denoting derivatives)

$$\kappa(\ell) = \frac{E[\ell'(e(t))]^2}{E\ell''(e(t))^2} \quad (41)$$

Note that

$$\kappa(\ell) = \sigma^2 = Ee^2(t) \quad \text{if } \ell(e) = e^2/2$$

If the model structure contains the true system, it can be shown that this covariance matrix is the smallest that can be achieved by any unbiased estimate, in case the norm ℓ is chosen as the logarithm of the pdf of e. That is, it fulfills the *the Cramér-Rao inequality* (Cramér 1946). These results are valid for quite general model structures. Now, specialize to linear models (5a) and assume that the true system is described by

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \quad (42)$$

which could be general transfer functions, possibly much more complicated than the model. Then

- $$\theta^* = \arg\min_{\theta} \int_{-\pi}^{\pi} |G(e^{i\omega}, \theta) - G_0(e^{i\omega})|^2$$
  $$\frac{\Phi_u(\omega)}{|H(e^{i\omega}, \theta)|^2} d\omega \quad (43)$$

That is, the frequency function of the limiting model will approximate the true frequency function as well as possible in a frequency norm given by the input spectrum Φ_u and the noise model.

- For a linear black-box model

$$\text{Cov}G(e^{i\omega}, \hat{\theta}_N) \sim \frac{n}{N} \frac{\Phi_v(\omega)}{\Phi_u(\omega)} \text{ as } n, N \to \infty \quad (44)$$

where n is the model order and Φ_v is the noise spectrum $\sigma^2|H_0(e^{i\omega})|^2$. The variance of the estimated frequency function at a given frequency is thus, for a high-order model, proportional to the noise-to-signal ratio at that frequency. That is a natural and intuitive result.

Trade-Off Between Bias and Variance

Generally speaking the quality of the model depends on the quality of the measured data and the flexibility of the chosen model structure (3). A more flexible model structure typically has smaller bias, since it is easier to come closer to the true system. At the same time, it will have a higher variance: With higher flexibility it is easier to be fooled by disturbances. So the trade-off between bias and variance to reach a small total error is a choice of balanced flexibility of the model structure.

As the model gets more flexible, the fit to the estimation data in (19), $V_N(\hat{\theta}_N)$, will always improve. To account for the variance contribution, it is thus necessary to modify this fit to assess the total quality of the model. A much used technique for this is Akaike's criterion, (AIC) (Akaike 1974):

$$\hat{\theta}_N = \arg\min_{\mathcal{M}, \theta \in D_{\mathcal{M}}} 2L_N(\theta) + 2\dim\theta \quad (45)$$

where L_N is the negative log likelihood function. The minimization also takes place over a family of model structures with different number of parameters (dim θ).

For Gaussian innovations e with unknown and estimated variance, AIC takes the form

$$\hat{\theta}_N = \underset{\mathcal{M}, \theta \in D_{\mathcal{M}}}{\arg\min} \left[\log \det \left[\frac{1}{N} \sum_{t=1}^{N} \varepsilon(t, \theta) \varepsilon^T(t, \theta) \right] + 2 \frac{\dim \theta}{N} \right] \tag{46}$$

after normalization and omission of model-independent quantities.

A variant of AIC is to put a higher penalty on the model complexity:

$$\hat{\theta}_N = \arg\min \left[2L_N(\theta) + \dim \theta \log N \right] \tag{47}$$

This is known as Bayesian information criterion (BIC) or Rissanen's minimum description length (MDL) criterion (Rissanen 1978).

Section "$\mathcal{V}$: Model Validation" contains further aspects on the choice of model structure.

$\mathcal{X}$: Experiment Design

Experiment design is the question of choosing which signal to measure, the sampling rate, and designing the input.

The theory of experiment design primarily relies upon analysis of how the asymptotic parameter covariance matrix (40) depends on the design variables: so the essence of experiment design can be symbolized as

$$\min_{\mathcal{X}} \text{trace}\{C[E\psi(t)\psi^T(t)]^{-1}\}$$

where ψ is the gradient of the prediction wrt the parameters and the matrix C is used to weight variables reflecting the intended use of the model.

For linear systems the input design is often expressed as selecting the spectrum (frequency contents) of u.

This leads to the following recipe: Let the input's power be concentrated to frequency regions where a good model fit is essential and where disturbances are dominating.

Issues of experiment design are treated in much more detail in ▸ Experiment Design and Identification for Control.

The measurement setup, like if band-limited inputs are used to estimate continuous-time models and how the experiment equipment is instrumented with band pass filters (see, e.g., Pintelon and Schoukens 2012, Sections 13.2–3), also belongs to the important experiment design questions.

$\mathcal{V}$: Model Validation

Model validation is about examining and scrutinizing an estimated model to check if it can be used for its purpose. These methods unavoidably are problem dependent and contain several subjective elements, and no conclusive procedure for validation can be given. A few useful techniques will be listed in this section. Basically it is a matter of trying to falsify a model under the conditions it will be used for and also to gain confidence in its ability to reproduce new data from the system.

Falsifying Models: Residual Analysis

An estimated model is never a correct description of a true system. In that sense, a model cannot be "validated." Instead it is instructive to try and *falsify* it, i.e., confront it with facts that may contradict its correctness. A good principle is to look for the *simplest unfalsified model*; see, e.g., Popper (1934).

Residual analysis is the leading technique for falsifying models: The residuals, or one-step-ahead prediction errors $\hat{\varepsilon}(t) = \varepsilon(t, \hat{\theta}_N) = y(t) - \hat{y}(t|\hat{\theta}_N)$ should ideally not contain any traces of past inputs or past residuals. If they did, it means that the predictions are not ideal. So, it is natural to test the correlation functions

$$\hat{r}_{\hat{\varepsilon},u}(k) = \frac{1}{N} \sum_{t=1}^{N} \hat{\varepsilon}(t+k)u(t) \qquad (48)$$

$$\hat{r}_{\hat{\varepsilon}}(k) = \frac{1}{N} \sum_{t=1}^{N} \hat{\varepsilon}(t+k)\hat{\varepsilon}(t) \qquad (49)$$

and check that they are not larger than certain thresholds. Here N is the length of the data record and k typically ranges over a fraction of the interval $[-N\,N]$. See, e.g., Ljung (1999), Section 16.6 for more details.

Comparing Different Models
When several models have been estimated, it is a question to choose the "best one." Then, models that employ more parameters naturally show a better fit to the data, and it is necessary to outweigh that. The model selection criteria AIC (46) and BIC (47) are examples of how such decisions can be taken. They can be extended to regular hypothesis tests where more complex models are accepted or rejected at various test levels (Ljung 1999, Sect. 16.4).

Making comparisons in the frequency domain is a very useful complement for domain experts who are used to think in terms of natural frequencies, natural damping, etc.

Cross Validation
Cross validation is an important statistical concept that loosely means that the model performance is tested on a data set (*validation data*) other than the estimation data. There is an extensive literature on cross validation, e.g., Stone (1977), and many ways to split up available data into estimation and validation parts have been suggested. A simple way, often used in system identification, is to use one-half of the data to estimate the model and the other half to evaluate simulation or prediction fit. Trying out different model structures (or other decision variables, like regularization parameters), one then picks the choice that gives the best performance on validation data.

Other Topics

Numerical Algorithms and Software Support
The central numerical task to estimate the model lies in the innocent-looking "arg min" in (38). Since the criterion often is non-convex, this global minimization can be nontrivial. Typically some iterative numerical optimization method, like Gauss-Newton, Levenberg-Marquardt, or trust regions, e.g., Nocedal and Wright (2012), is employed. The iterations are initiated at a carefully selected point, for black-box linear systems often based on ARX or subspace estimates.

The practical use of system identification relies upon efficient software support. Many such packages exist. They are further treated along with numerical and computational aspects in ▶ System Identification Software.

Estimating Continuous-Time Models
Most of the techniques described here formally seem to deal with estimating discrete time model. However continuous-time (CT) models are to be preferred in many contexts, and most of the modeling of physical systems really concern CT models. A natural approach is to do physical modeling in continuous time as in (11) and then do estimation of the CT matrices via the sampled model (12). All the described algorithms and results apply to this approach to CT model estimation. Another approach is to use band-limited inputs and compute the CT Fourier transforms of data (that coincide with the discrete time transforms for band-limited data) and apply ▶ Frequency Domain System Identification.

Yet another approach is to directly fit CT model parameters to discrete time data, using specially designed filters; see, e.g., Garnier and Wang (2008).

Recursive Estimation
For certain adaptive and in-line applications, it may be necessary to continuously compute the models by recursively updating the estimates. The techniques for that resemble state-estimation

algorithms and are dealt with in a general setting in ▶ Nonlinear System Identification Using Particle Filters. See also Ch 11 in Ljung (1999).

Data Management

The collected data often requires particular attention before it can be used for estimation. Issues like missing observations, obviously erroneous values (outliers), slowly varying disturbances, trends, etc., need attention. In industrial applications, a practical question is often to select portions of the data records that contain relevant information for the model building. Such questions are application dependent and related to experiment design and also to database management. Some techniques for preparing data for identification are mentioned in Ch 14 of Ljung (1999).

Summary and Future Directions

System identification is a mature and well-established area in automatic control. The methods are successfully and routinely applied in industrial practice, and the understanding of theoretical issues is mostly excellent. The standard theory relies very much on basic statistical concepts and methods.

What is exciting about future development is what increased computation power may mean for the area: Can nonlinear models be efficiently estimated by massive computational efforts? Will tools inspired by machine learning turn out to be superior to the conventional approaches? Can reliable uncertainty regions be computed for arbitrary noises and without the asymptotic formulas?

Several of these questions are illuminated in the articles listed under Cross-References.

Cross-References

There are several articles in this encyclopedia that deal with aspects of system identification. They have been coordinated with this overview and the text has listed how they complement the

issues treated here. For easy reference, here is a complete list of associated articles:

- ▶ Experiment Design and Identification for Control
- ▶ Frequency Domain System Identification
- ▶ Modeling of Dynamic Systems from First Principles
- ▶ Nonlinear System Identification: An Overview of Common Approaches
- ▶ Nonlinear System Identification Using Particle Filters
- ▶ Nonparametric Techniques in System Identification
- ▶ Subspace Techniques in System Identification
- ▶ System Identification Software
- ▶ System Identification Techniques: Convexification, Regularization, and Relaxation

Recommended Reading

A text book that covers and extends the material in this contribution is Ljung (1999). Another text book in the same spirit is Söderström and Stoica (1989), while Pintelon and Schoukens (2012) gives a comprehensive treatment of frequency domain methods. Recursive methods are treated in Young (2011).

Bibliography

Akaike H (1974) A new look at the statistical model identification. IEEE Trans Autom Control AC-19:716–723

Cramér H (1946) Mathematical methods of statistics. Princeton University Press, Princeton

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithms. J R Stat Soc Ser B 39(1):1–38

Garnier H, Wang L (eds) (2008) Identification of continuous-time models from sampled data. Springer, London

Ljung L (1999) System identification – theory for the user, 2nd edn. Prentice-Hall, Upper Saddle River

Ljung L (2002) Prediction error estimation methods. Circuits Syst Signal Process 21(1):11–21

Nocedal J, Wright J (2012) Numerical optimization, 2nd edn. Springer, Berlin

Pillonetti G, Dinuzzo F, Chen T, De Nicolao G, Ljung L (2014) Kernel methods in system identification, machine learning and function estimation: a survey. Automatica 50(3):657-683

Pintelon R, Schoukens J (2012) System identification – a frequency domain approach, 2nd edn. IEEE, New York

Popper KR (1934) The logic of scientific discovery. Basic Books, New York

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. MIT, Cambridge

Rissanen J (1978) Modelling by shortest data description. Automatica 14:465–471

Söderström T (2007) Errors-in-variables identification in system identification. Automatica 43(6): 939–958

Söderström T, Stoica P (1983) Instrumental variable methods for system identification. Springer, New York

Söderström T, Stoica P (1989) System identification. Prentice-Hall, London

Stone M (1977) Asymptotics for and against cross-validation. Biometrica 64(1):29–35

Tikhonov AN, Arsenin VY (1977) Solutions of Ill-posed problems. Winston/Wiley, Washington, DC

Young PC (2011) Recursive estimation and time-series analysis, 2nd edn. Springer, Berlin

Zadeh LA (1956) On the identification problem. IRE Trans Circuit Theory 3:277–281

S

T

Tactical Missile Autopilots

Curtis P. Mracek
Raytheon Missile Systems, Waltham, MA, USA

Abstract

Tactical missile autopilots are part of the wider guidance navigation and control missile system whose goal is to achieve a successful intercept. The missile autopilot task is to turn guidance commands into fin deflection and is generally divided into two lateral direction (pitch and yaw) controllers and the roll orientation or roll rate controller. These three "channel control" outputs are then mixed to produce fin commands. The controllers can be composed of different architectures but most lateral autopilots use a three loop structure with acceleration and angular rate feedback. The roll controller is usually either a proportional integral (PI) or proportional integral derivative (PID) controller. The controllers are designed using gain scheduling for large flight envelope applications and have nonlinear elements to shape the time response. Integrator reset logic, to deal with control surface saturation, is also an integral part of tactical missile autopilots.

Keywords

Classical control; Control surfaces; Pitch; Proportional and integral control; Roll channel; Tactical missile; Yaw channels

Introduction

The purpose of a tactical missile is to intercept targets, and since tactical missile autopilots are part of the larger tactical missile system, they must contribute to that goal. The process by which a missile executes an intercept is by first sensing the target. The target information is then used to generate guidance commands. The guidance commands are determined such that if followed with precision the missile will intercept the target. The problem is to follow with precision. This is where the autopilot comes in. The missile autopilot receives guidance commands and produces control deflections to move the missile in a manner consistent with completing the intercept. There are many control challenges unique to tactical missiles, namely closing velocities can be very high and targets very small and very maneuverable. Usually the guidance commands are acceleration commands though other quantities are sometimes used. For this discussion, acceleration commands will be the autopilot

commands. Once the acceleration commands or demands (as some in the guidance community call the autopilot inputs) are presented to the autopilot, the autopilot's only concern is to produce the desired command as fast as possible with some level of robustness. The key performance metric is the time of response. The response time is a key factor that drives the miss distance, and thus the probability of a successful intercept. Another metric, though less important than the response time, is the available maneuverability. As mentioned, the autopilot achieves the desired acceleration through moving the control surfaces. Usually, the control surfaces are aerodynamic and either positioned in front of (canard control) or in back of (tail control) the center of gravity. Both tail and canard control surfaces will be called fins for the purposes of this analysis. Some recent missile designs have significantly altered the autopilot design problem by using both canards and tails or other effectors like reaction jets.

The tactical missile autopilot control problem is therefore to produce accelerations by moving the fins in a controlled manner such that the response is as fast as possible while remaining under control under various flight conditions and in the presence of uncertainties (being robust). Tactical missiles autopilots are a classic control challenge in that there is a direct trade between performance and robustness. The tactical autopilot tends to lean toward the performance instead of the robustness because, as the continuing argument goes, "what good is the missile being stable if you miss the target" versus "if the missile is unstable it may never get to the target." So far, relevant analysis has mentioned the controller but mostly ignored robustness. Achieving robustness is done through the use of feedback, and in tactical missiles, inertial sensing devices are used to provide this critical information. These tactical sensors are currently packaged as a complete inertial sensor suite. This suite usually consists of three orthogonal linear accelerometers and three angular rate gyros. One reason guidance commands are the linear acceleration is that the sensing device directly measures this desired quantity.

There are two noticeable differences between tactical missile control and other aerodynamic control applications. The first is that the dynamics and controls are divided into three distinct channels with each channel nearly independent of the other two. These are the lateral (pitch and yaw) channels and the axial (roll) channel. The pitch and yaw designs are usually very similar, if not identical, and the roll channel is separate. The second is that the controllers (fins) are intertwined. That is, there are no predominately pitch, yaw, or roll controllers, such as there are on airplanes. At least two and sometimes four fins are used in a single channel. This mixing of controls is through what is called a fin mix. This fin mixing occurs in the software (used to be hardware in analog controllers) after the autopilot and prior to the signals being sent to the individual fins.

Historically tactical missile autopilot development has consisted of both a design phase and an analysis phase. This distinction is due to the controller being designed on a subset of the operating envelope. That design is then evaluated at many more conditions to determine if the design works well enough everywhere to be deployed. In both the design and analysis phases, models are used to establish performance and robustness. Linear planar, linear coupled, and nonlinear models are used. The linear models are usually restricted to the early design phases and the frequency response determination of the system. The nonlinear models are used for time domain analysis.

The remainder of the chapter is organized by examining the linear planar pitch and yaw autopilots, followed by roll control. The concept of combining controllers is then presented. This is followed by a short section on other considerations, such as coupled designs and nonlinear elements.

Pitch and Yaw Control

For tactical missile autopilot development, the equations of motion are usually derived in a body-fixed system with the two lateral velocities replaced by the local angle of attack (α) and sideslip angle (β). It should be noted that the

sideslip is not defined as the aircraft sideslip but instead as the equivalent of the angle of attack in the horizontal plane. This is because of the symmetry that is found in missiles that does not exist in aircraft. The nonlinear equations of motion can be found in Blakelock (1991). For tactical missiles there is usually no axial acceleration control, and thus the total velocity equation is uncontrollable and removed from both the design and analysis. For the coupled equations of motion of the system, there are five equations of motion and three control inputs. For a planar view of the problem, the pitch and yaw channels in a tactical missile autopilot are usually separated, and with the appropriate sign changes in the feedback signals can use the same gains. These channels use an inertial measuring device for feedback. Usually these sensors come in a package with three accelerometers and the gyros. The outputs of these devices used in the pitch are the linear acceleration perpendicular to the axial direction and the angular rate of the missile about the other perpendicular axis. That is, the z linear acceleration (A_{zm}) and the y angular rate (q_m). Using the other four sensors would cause coupling between pitch and yaw and roll, and thus this sensor information is usually ignored in the pitch channel. They are available and used in select cases where there is strong aerodynamic coupling, in which case these cross channels can be used to decouple the system. Without getting into the actual definitions of all the variables and the numerical values (see Mracek and Ridgely 2005a for full details), the state space linearized equations of motion for the pitch plane are:

$$A = \begin{bmatrix} 1/V_{mo}\left[\frac{Z_{\alpha o}}{m} - A_{Xo}\right] & 1 \\ M_{\alpha o}/I_{YY} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} Z_{\delta po}/mV_{mo} \\ M_{\delta po}/I_{YY} \end{bmatrix}$$

$$C = \begin{bmatrix} Z_{\alpha o}/mg - M_{\alpha o}\bar{x}/gI_{YY} & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} Z_{\delta po}/mg - M_{\delta po}\bar{x}/gI_{YY} \\ 0 \end{bmatrix}$$

where

$$\dot{x} = Ax + Bu \quad x = \begin{bmatrix} \alpha \\ q \end{bmatrix} \quad u = \delta_p \quad y = \begin{bmatrix} A_{zm} \\ q_m \end{bmatrix}$$
$$y = Cx + Du$$

Thus in the most reduced form, the tactical missile autopilot equations of motion reduce to two equations with two variables and one control. This is a very simple control problem. Since full state feedback can be used to provide an "optimal" control solution, only two feedback signals are needed for the above state space problem. For a tail controlled missile, the two state control leads ultimately to increasing missile acceleration in the wrong direction, as faster and faster designs are realized. This is because the system is, in controls language, "non-minimum phase." That is, tail controlled missiles move in the wrong direction before they move in the commanded direction. Canard controlled missiles do not suffer this problem. See Mracek (2005) and Gutman (2003) on the relative merits of canards and tails. If the control rate is used as the input instead of the control position, there would be three states in the basic plant used in the analysis, and three signals would need to be included. Now if we consider the fin position as a variable for feedback with the accelerometer and gyro feedbacks, there are a number of different combinations of sensor feedback signals that can be used to solve the three state problem. There are, in fact, nine possible topologies, two of which are consistently robust, with one topology showing excellent robustness characteristic. For a complete comparison see Mracek and Ridgely (2005b). This topology is shown in Fig. 1. Notice that there is an integral in the formulation. This limits the actual command rate from being infinite when a step command is input to the system. Without the command going through the integrator the controller would see the step, and, since the force instantly produces an acceleration, the feedback would jump (given no actuation delay). A typical acceleration response to a step acceleration command and control deflection rate needed to produce the response is presented in Figs. 2 and 3, respectively.
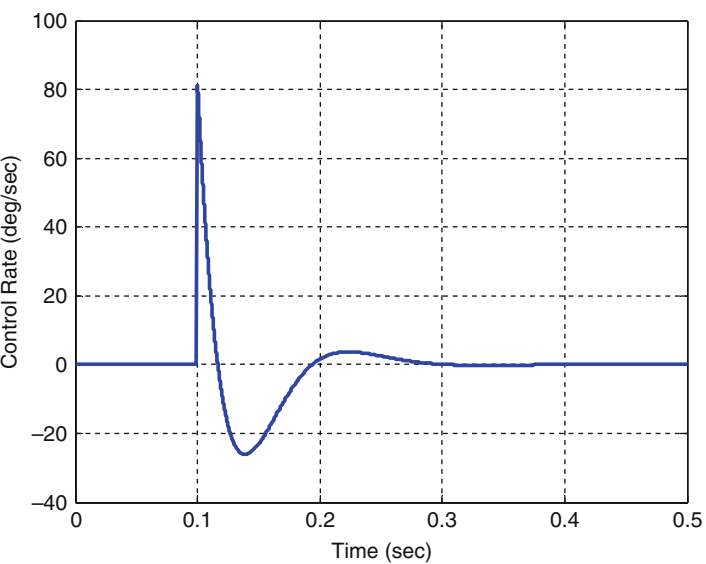
Tactical Missile Autopilots, Fig. 1 Three loop pitch topology
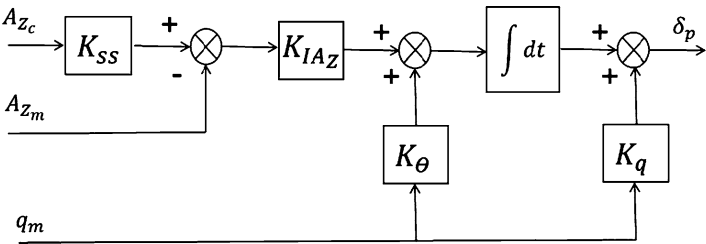
Tactical Missile Autopilots, Fig. 2 Acceleration response to a step input
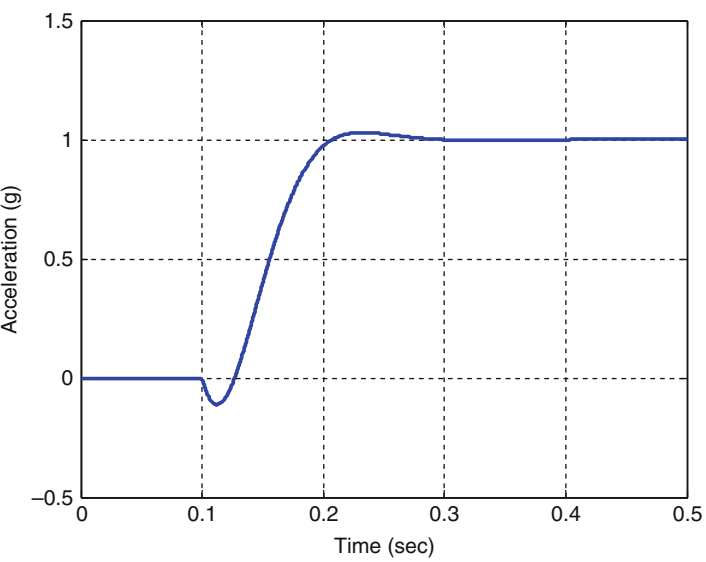
Tactical Missile Autopilots, Fig. 3 Control rate usage

The feedback control law is:

$$\delta_p = K_{IA_z}K_{ss} \int A_{Z_c} dt - K_{IA_z} \int A_{Z_m} dt$$

$$+ K_\theta \int q_m dt + K_q qm$$

Clearly, other components within the autopilot loop have to be considered. The control actuation system (CAS) and inertial measurement device characteristics need to be included in the design and synthesis of the autopilot. To this end, the gains are usually selected to provide the best performance (in the time domain) based on constraints. The above optimal control solution provides guaranteed margins, but when the additional components are included in the analysis the margins are an important constraint in the ultimate performance that can be achieved. Like most control problems, the constraints are both time and frequency dependent. Because of the emphasis on performance, some of the margin constraints must be examined closely. For a more detailed treatment of the three loop autopilot see Zarchan (2002).

Roll Control

Thus far we have discussed the two lateral channels of the missile. That is because those two are the channels that directly affect the miss distance. The third channel does not directly influence the miss but it still is usually controlled. The roll channel is usually the fastest of the channels for a tactical missile. Historically, the three channels were decoupled by moving the roll "out of the

way" of the other channels by designing to a higher bandwidth than the pitch and yaw channels. Because of the need for squeezing performance this practice is not always employed. The cost of not increasing the bandwidth of the roll beyond the pitch and yaw is that the interdependence of the channels needs more scrutiny. The roll channel has only one sensor element, the roll rate senor. This measures the angular rate of the body about its central axis relative to the inertial frame. The objective, and thus the autopilot, can differ depending on the missile application. Mostly the objective would be one of the following: maintain zero roll rate, zero integral of roll rate, or some preferred Euler angle orientation. The last two can be accomplished with the same autopilot architecture, with exception handling for the Euler roll control based on the singularity in the Euler roll angle at ±90° pitch orientations.

Since there is only one sensor and one control, this channel is a classical SISO system and can be controlled with a proportional derivative (PD) or proportional integral derivative (PID) controller. The three loop topologies with integral roll rate reference are presented in Fig. 4.

Gain Scheduling

Early generation missiles had analog autopilots and some were marvels of ingenuity. Now digital control is used almost exclusively. As can be readily seen from the above discussion, the autopilots performance is largely dictated by gains within a given topology. Unlike with early autopilots, with digital control the gains can be set precisely and can vary greatly as needed

Tactical Missile Autopilots, Fig. 4 Three loop roll topology
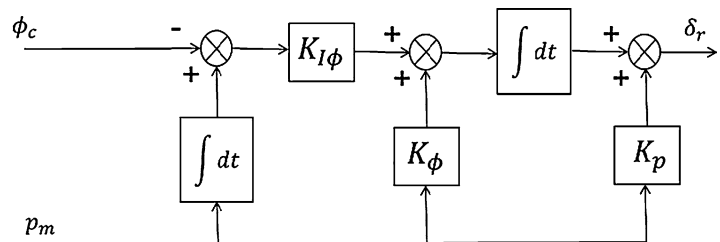
over a wide range of flight conditions. Rarely can a single set of gains be found that provides adequate performance under all conditions. Thus, the autopilot design process is to design for a large number of flight conditions and then join the individual designs into a coherent whole. Typically, the conditions for the individual designs would be something like Mach number, altitude, and center of mass location. Once the individual gains are designed, they are joined together through an algorithm. Most likely they are "looked up" as a continuous function of the independent variables through some sort of interpolation. This gain changing philosophy is called gain scheduling. There have been some successful attempts for full envelope autopilot design. Dynamic inversion or model-based approaches have also been developed, most notably JDAM (Wise et al. 2005) where the autopilot was borrowed and adjusted on the fly from a sister design. The argument for the validity of this approach is that the flight conditions are not changing rapidly so they can be ignored. Of course the synthesis of the design needs to include examination of "off break point" conditions (flight conditions within the flight envelope that were not considered in the design process) to ensure compliance with stability requirements. History has shown that tactical missile autopilot gains tend to be somewhat power functions of dynamic pressure based on the design constraints.

Other Considerations

The selection of gains using planar linear models and then scheduling them is not the complete autopilot design exercise. There are other challenges that must be considered. First, the plant equations are coupled through both the kinematic equations and the aerodynamics of the problem. There are two predominant ways to attack this problem in autopilot design. The first is as discussed earlier in which the system is made to be as decoupled as possible, create gains for the decoupled system and analyze them in the coupled system. The second is to use feedback to create a more integrated design through cross coupling terms.

Besides the coupling there can be other problems. The design problem is hard enough as described above, but we have learned over the years that the models developed earlier have neglected certain aspects of the missile that can lead to problems. One aspect is that missiles can be very flexible, and since an inertial sensor is being used for feedback, the flexible characteristics can drive the missile unstable. The flexible characteristics were examined by Nesline and Nesline (1985). In that paper, the flexible model is presented and a technique for ignoring the first mode is discussed. (It should be noted that the model presented in the appendix has some "typos" and should be used with caution.)

Another aspect is the consideration of nonlinear elements of the autopilot. These could include integrator reset logic, command error limits, and acceleration limits. The three loop autopilot has an integrator and the fact that integrators "wind up" when the output is saturated. For tactical missiles this saturation could be caused by position or rate limits. The integrator should be reset to account for these conditions so that the missile responds quicker when the system is no longer in a saturated condition. The command error limits can be used to modify the response characteristics to achieve a more consistent response. Finally, acceleration limits are used to limit the input into the system such that the guidance commands do not put the missile into a position from which it cannot maintain controlled flight. Generating acceleration limits is a complex topic itself.

Summary and Future Directions

Tactical missile autopilots are generally designed by separating the problem into two independent

lateral controls (pitch and yaw), with a third control governing the roll attitude. A good autopilot design produces a balance between performance and robustness and incorporates nonlinear elements and integrator resets. The design process take into account robustness throughout the flight envelope and structural elements.

From a controls standpoint, the future direction of tactical missile autopilot development is in nonlinear, adaptive, and fault tolerant control. Adaptive control is useful not only because it provides a more predictable flight response but also because of the potential in reducing or maybe even eliminating development time.

Cross-References

▶ Aircraft Flight Control
▶ PID Control

Bibliography

Blakelock JH (1991) Automatic control of aircraft and missiles, 2nd edn. Wiley, New York
Gutman S (2003) Superiority of canards in homing missiles. IEEE Trans Aerosp Electron Syst 39(3): 740–746
Mracek CP (2005) A miss distance study for homing missiles: tail vs canard control. In: Proceedings of the AIAA guidance navigation and control conference, Minneapolis, Aug 2006
Mracek CP, Ridgely DB (2005a) Missile longitudinal autopilots: connections between optimal control and classical topologies. In: Proceedings of the AIAA GNC conference, San Francisco, Aug 2005
Mracek CP, Ridgely DB (2005b) Missile longitudinal autopilots: comparison of multiple three loop topologies. In: Proceedings of the AIAA guidance navigation and control conference, San Francisco, Aug 2005
Nesline FW, Nesline ML (1985) Phase vs gain stabilization of structural feedback oscillations in homing missiles. In: Proceedings of the American control conference, 1985
Wise KA, Lavretsy E, Zimmerman J, Francis JH Jr, Dixon D, Whitehead B (2005) Adaptive flight control of a sensor guided munition. In: Proceedings of the AIAA guidance navigation and control conference, San Francisco, Aug 2005
Zarchan P (2002) Tactical and strategic missile guidance. Progress in Astronautics and Aeronautics, vol 199, 4th edn. AIAA, Reston

Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency

Joe H. Chow
Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract

Large power systems often exhibit slow and fast electromechanical oscillations between interconnected synchronous machines. The slow interarea oscillations involve coherent groups of machines swinging together. This coherency phenomenon can be attributed to the coherent areas of machines being weakly coupled, either because of higher impedance transmission lines, heavily loaded transmission lines, or fewer connections between the coherent areas compared to the connections within a coherent area. Singular perturbations can be used to display the time-scale separation of the slow interarea modes and the faster local modes.

Keywords

Model reduction; Power system oscillations; Singular perturbations; Two-time-scale systems

Interarea Mode Oscillation in a Power System

A large power system consists of interconnected synchronous machines supplying power to loads via transmission lines. As a dynamical system, it can be considered as the rotating inertias of the synchronous machines interacting electrically through the impedances of the transmission system. During a disturbance, such as a lightning strike on a transmission line, the rotating inertias will oscillate against each other. The frequency and extent of these oscillations
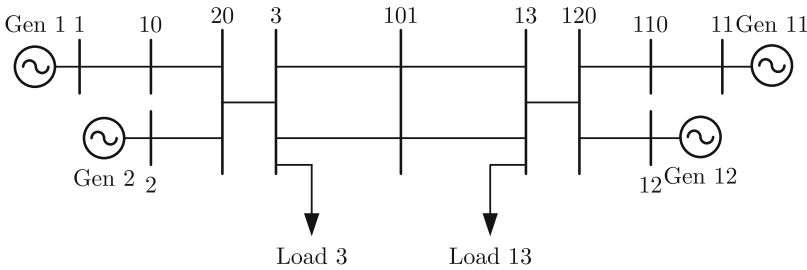
Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency, Fig. 1 Two-area, four-machine system example

Time-Scale Separation in Power System Swing Dynamics: Singular Perturbations and Coherency, Fig. 2 Machine speed response of the two-area, four-machine system
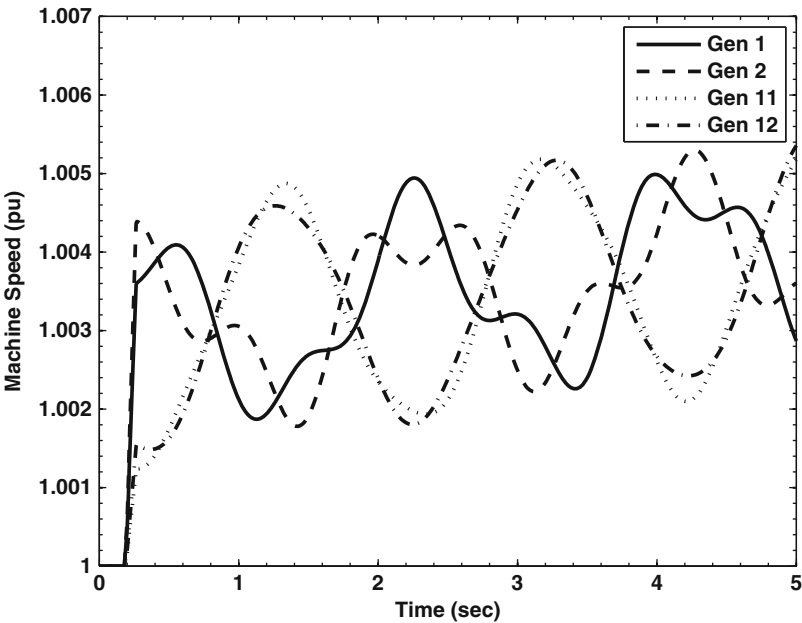
may vary: the local modes of frequencies 1–2.5 Hz originate from the interactions of a few close-by machines, and the interarea modes of frequencies 0.2–0.8 Hz involve groups of machines swinging against other groups. Coherency is this phenomenon of groups of machines swinging together against other groups of machines during disturbances.

Coherency can be illustrated in the simple power system shown in Fig. 1 (Rogers 2000). The system consists of two areas: Generators 1 and 2 in Area 1 and Generators 11 and 12 in Area 2. For a disturbance in Area 1, Fig. 2 shows the response of the machine speeds. The interarea mode consists of Generators 1 and 2 swinging coherently against Generators 11 and 12. The

difference between the responses of Generators 1 and 2 is due to the local mode in Area 1, which is excited by the disturbance.

Coherency Analysis

Coherency with respect to the slow interarea modes, also known as slow coherency, is an inherent property of many power systems. Traditional power systems consist of operating regions dictated by physical or administrative constraints with relatively strong connections within an operating region. These control regions are also interconnected with tielines to share base-load and seasonal power resources as well as to rely on

each other for reserves. Thus a practical interconnected power system will, by design, necessarily have strong connections within each operating region and weaker connections between the regions. Due to the time-scale separation of the slow interarea modes and the faster local modes, the coherency phenomenon can be analyzed using singular perturbations method provided a suitable small parameter can be identified.

For a simplified coherency analysis, the linearized second-order model of an N-machine power system

$$M \frac{d^2 \Delta \delta}{dt^2} = K \Delta \delta \qquad (1)$$

can be used. In (1), δ is the N-dimensional vector of individual machine rotor angles δ_i, $i = 1, \ldots, N$, Δ denotes small perturbations, M is the diagonal matrix of machine rotational inertias m_i, $i = 1, \ldots, N$, and the *connection matrix K* consists of the linearized synchronizing coefficients K_{ij} between machines i and j, denoting the restoring force between the two machines.

An important property of K is

$$K_{ii} = - \sum_{j=1, j \neq i}^{N} K_{ij} \qquad (2)$$

that is, the sum of each row of K is zero. Thus K has a zero eigenvalue, which is known as the system mode. This mode arises due to the lack of a reference, as only the relative angles between the machines are important. It can be eliminated when one of the machines is chosen as the reference.

Suppose the N-machine system has r areas of coherent machines, whose internal connections within the areas are stronger than the external connections between the areas. The weak connection strength is denoted by a small parameter ε, which can be the ratio of the relative stiffness of the internal transmission lines versus the external transmission lines, or the ratio of the smaller number of external connections versus the larger number of internal connections, or both. Thus the connection matrix of linearized synchronizing coefficients can be rewritten as

$$K = K^I + \varepsilon K^E \qquad (3)$$

where K^I is the matrix of internal connections and K^E is the matrix of external connections scaled by ε. If the machine angles in each coherent area are arranged in consecutive order in the vector δ, then K^I is block diagonal with r zero eigenvalues, that is, one system mode per area.

Singular Perturbation Analysis

To exhibit the time scales in (1) and (3), a transformation to obtain the slow variables and the fast variables is introduced. The slow motion is obtained by defining for each area, an inertia-weighted *aggregate variable*

$$y^\alpha = \sum_{i=1}^{n_\alpha} m_i^\alpha \Delta \delta_i^\alpha / m^\alpha,$$

$$m^\alpha = \sum_{i=1}^{n_\alpha} m_i^\alpha, \quad \alpha = 1, 2, \ldots, r \qquad (4)$$

where n_α is the number of machines in area α, m_i^α is the inertia of machine i in area α, and m^α is the aggregate inertia of area α. For the fast dynamics, we select in each area a reference machine, say the first machine, and define the motions of the other machines in the same area relative to this reference machine by the *local variables*

$$z_{i-1}^\alpha = \Delta \delta_i^\alpha - \Delta \delta_1^\alpha, \quad i = 2, 3, \ldots, n_\alpha,$$

$$\alpha = 1, 2, \ldots, r \qquad (5)$$

The transformations (4) and (5) can be combined to form

$$\begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} M_a^{-1} U^T M \\ G \end{bmatrix} \Delta \delta \qquad (6)$$

where

$$U = \text{blockdiag}(u_1, u_2, \ldots, u_r) \qquad (7)$$

is the grouping matrix with $n_\alpha \times 1$ column vectors

$$u_\alpha = \begin{bmatrix} 1 \ 1 \ \ldots \ 1 \end{bmatrix}^T, \quad \alpha = 1, 2, \ldots, r \quad (8)$$

$$M_a = \mathrm{diag}(m^1, m^2, \ldots, m^r) = U^T M U \quad (9)$$

and

$$G = \mathrm{blockdiag}(G_1, G_2, \ldots, G_r) \quad (10)$$

with G_α being the $(n_\alpha - 1) \times n_\alpha$ matrix

$$G_\alpha = \begin{bmatrix} -1 & 1 & 0 & . & 0 \\ -1 & 0 & 1 & . & 0 \\ . & . & . & . & . \\ -1 & 0 & 0 & . & 1 \end{bmatrix} \quad (11)$$

The inverse of this transformation is explicitly known

$$\Delta\delta = \begin{bmatrix} U \ G^T(GG^T)^{-1} \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \quad (12)$$

Applying the transformation (6) to the model (1) and (3), the electromechanical model becomes

$$M_a\ddot{y} = \varepsilon K_a y + \varepsilon K_{ad} z$$
$$M_d\ddot{z} = \varepsilon K_{da} y + (K_d + \varepsilon K_{dd}) z \quad (13)$$

where

$$M_d = (GM^{-1}G^T)^{-1}, \quad K_a = U^T K^E U$$

$$K_{da} = U^T K^E M^{-1} G^T M_d,$$

$$K_{da} = M_d G M^{-1} K^E U$$

$$K_d = M_d G M^{-1} K^I M^{-1} G^T M_d,$$

$$K_{dd} = M_d G M^{-1} K^E M^{-1} G^T M_d \quad (14)$$

Note that K_a, K_{ad}, and K_{da} are independent of the internal connection matrix K^I because $K^I U = 0$. Furthermore, K_a is negative semi-definite and K_{dd} is negative definite. System (13) is in the *standard singularly perturbed form* (Kokotović et al. 1986) showing that y is the slow variable and z is the fast variable. Thus ε is both the weak connection parameter and the singular perturbation parameter, giving rise to slow coherency.

The dynamics of the singularly perturbed system (13) are approximated by the interarea modes $\pm j\sqrt{-\varepsilon\lambda(M^{-1}K_a)}$ and the local modes $\pm j\sqrt{-\lambda(M_d^{-1}K_{dd})}$, where λ denotes eigenvalues.

Identifying Coherent Areas

Several methods can be used to identify coherent areas, including the following:

1. Time simulation method (Podmore 1978): This method simulates the dynamic responses to a selected set of disturbances and groups the machines having similar time responses as coherent areas. For a faster simulation, a linearized power system model can be used.
2. Eigenvector method (Chow et al. 1982): This method computes the slow eigenvalues of the matrix $M^{-1}K$ and identifies machines with similar row vectors of the slow eigenvector matrix as coherent machines.
3. Weak link methods (Nath et al. 1985; Zaborszky et al. 1982): These methods search through the transmission line impedances to find the weak links between the areas.

Applications

The applications of the coherency concept include:

1. Dynamic model reduction (deMello et al. 1975): The synchronous machines in a coherent area can be aggregated into a single equivalent machine, thus reducing the system size. Model reduction programs capable of handling upwards of 30,000 buses are available (Morison and Wang 2013).
2. Interarea mode analysis and damping control design (Larsen et al. 1995): Damping of interarea modes is an operational concern for systems with heavily loaded long-distance transmission lines. The slow coherency concept contributes to the development of damping controller design.

3. Islanding as a defense mechanism (You et al. 2004): During system disturbances causing severe power flow interruption, the last resort may be to separate the systems into viable islands, avoiding a total system blackout. Coherent areas tend to be natural choices of islands.

In addition to power system analysis, the coherency concept and methods can potentially be applied to dynamic systems with a system mode (eigenvalue equal to 0 for a continuous-time model and eigenvalue equal to 1 for a discrete-time model). An example is the PageRank computation in (Ishii et al. 2012).

Cross-References

▶ Consensus of Complex Multi-agent Systems
▶ Lyapunov Methods in Power System Stability
▶ Markov Chains and Ranking Problems in Web Search
▶ Model Order Reduction: Techniques and Tools
▶ Small Signal Stability in Electric Power Systems

Recommended Reading

An early investigation of coherency was reported in Podmore and Germond (1977). A recent compilation of power system coherency, model reduction, and interarea oscillation results can be found in Chow (2013).

Bibliography

Chow JH (ed) (2013) Power system coherency and model reduction. Springer, New York

Chow JH, Peponides G, Kokotović PV, Avramović B, Winkelman JR (1982) Time-scale modeling of dynamic networks with applications to power systems. Springer, New York

deMello RW, Podmore R, Stanton KN (1975) Coherency-based dynamic equivalents: applications in transient stability studies. 1975 PICA conference proceedings, pp 23–31

Ishii H, Tempo R, Bai E-W (2012) A web aggregation approach for distributed randomized PageRank algorithms. IEEE Trans Autom Control 57:2703–2717

Kokotović PV, Khalil H, O'Reilly J (1986) Singular perturbation methods in control: analysis and design. Academic, London

Larsen EV, Sanchez-Gasca JJ, Chow JH (1995) Concepts for design of FACTS controllers to damp power swings. IEEE Trans Power Syst 10:948–956

Morison K, Wang L (2013) Reduction of large power system models: a case study. In: Chow JH (ed) Power system coherency and model reduction. Springer, New York, Chapter 7

Nath R, Lamba SS, Rao KSP (1985) Coherency based system decomposition into study and external areas using weak coupling. IEEE Trans Power Appar Syst PAS-104:1443–1449

Podmore R, Germond A (1977) Dynamic equivalents for transient stability studies. EPRI Report RP-765

Podmore R (1978) Identification of coherent generators for dynamic equivalents. IEEE Trans Power Appar Syst PAS-97(4):1344–1354

Rogers G (2000) Power system oscillations. Kluwer Academic, Dordrecht

You H, Vittal V, Wang X (2004) Slow coherency-based islanding. IEEE Trans Power Syst 19: 483–491

Zaborszky J, Whang K-W, Huang GM, Chiang L-J, Lin S-Y (1982) A clustered dynamical model for a class of linear autonomous systems using simple enumerative sorting. IEEE Trans Circuit Syst CAS-29:747–758

Tracking and Regulation in Linear Systems

A. Astolfi
Department of Electrical and Electronic Engineering, Imperial College London, London, UK
Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma Tor Vergata, Roma, Italy

T

Abstract

Tracking and regulation refer to the ability of a control system to track/reject a given family of reference/disturbance signals modelled as solutions of a differential/difference equation. The problem can be posed as a stabilization problem with a constraint on the steady-state response of the system. For linear, time-invariant, systems, the problem can be solved provided a system of linear matrix equations admits a

solution. Properties of this system of equations are discussed, together with a general property of all controllers achieving tracking and regulation: the so-called internal model principle.

Keywords

Internal model principle; Linear systems; Regulation; Tracking

Introduction

Consider a linear system affected by disturbances and such that its output is required to asymptotically track a certain, prespecified, reference signal. In what follows, we discuss and solve this control problem known as the *tracking and regulation problem*.

Consider a linear control system described by equations of the form

$$\sigma x = Ax + Bu + Pd,$$
$$e = Cx + Qd, \tag{1}$$

with $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $e(t) \in \mathbb{R}^p$, $d(t) \in \mathbb{R}^r$, and A, B, P, C, and Q constant matrices. In Eq. (1), $\sigma x = \sigma x(t)$ stands for $\dot{x}(t)$, if the system is continuous-time, and for $x(t + 1)$, if the system is discrete-time. Since the system is time-invariant, it is assumed, without loss of generality, that all signals are defined for $t \geq 0$, that is, if the system is continuous-time, then $t \in \mathbb{R}^+$, i.e., the set of nonnegative real numbers, whereas if the system is discrete-time, then $t \in \mathbb{Z}^+$, i.e., the set of nonnegative integers. For ease of notation, the argument "t" is dropped whenever this does not cause confusion, and we use the notation $t \geq 0$ to denote either $\mathbb{R}^+$ or $\mathbb{Z}^+$.

The signal $d(t)$, denoted exogenous signal, is in general composed of two components: the former models a set of disturbances acting on the system to be controlled and the latter a set of reference signals. In what follows we assume that the exogenous signal is generated by a linear system, denoted exosystem, described by the equation

$$\sigma d = Sd, \tag{2}$$

with S a matrix with constant entries. Note that, under this assumption, it is possible to generate, for example, constant or polynomial references/disturbances and sinusoidal references/disturbances with any given frequency.

The variable $e(t)$, denoted tracking error, is a measure of the error between the ideal behavior of the system and the actual behavior. Ideally, the variable $e(t)$ should be regulated to zero, i.e., should converge asymptotically to zero, despite the presence of the disturbances. If this happens, we say that the tracking error is regulated to zero, i.e., converges asymptotically to zero; hence, the disturbances are not affecting the asymptotic behavior of the system and the output $Cx(t)$ is asymptotically tracking the reference signal $-Qd(t)$. In general the tracking error does not naturally converge to zero; hence, it is necessary to determine an input signal $u(t)$ which *drives* it to zero. The simplest possible way to construct such an input signal is to assume that it is generated via static feedback of the state $x(t)$ of the system to be controlled and of the state $d(t)$ of the exosystem, i.e.,

$$u = Kx + Ld. \tag{3}$$

In practice it is unrealistic to assume that both $x(t)$ and $d(t)$ are measurable; hence, it may be more natural to assume that the input signal $u(t)$ is generated via dynamic feedback of the error signal only, i.e., it is generated by the system

$$\sigma\chi = F\chi + Ge$$
$$u = H\chi, \tag{4}$$

with $\chi(t) \in \mathbb{R}^\nu$, for some $\nu > 0$, and F, G, and H matrices with constant entries.

Using the above definitions, it is possible to formally pose the regulator problem as follows.

Definition 1 (Full information regulator problem) Consider the system (1), driven by the exosystem (2) and interconnected with the controller (3). The full information regulator problem is the problem of determining the matrices K

and L of the controller such that ((S) stands for stability and (R) for regulation):

(S) The system $\sigma x = (A + BK)x$ is asymptotically stable.

(R) All trajectories of the system

$$
\begin{aligned}
\sigma d &= Sd, \\
\sigma x &= (A + BK)x + (BL + P)d, \quad (5) \\
e &= Cx + Qd,
\end{aligned}
$$

are such that $\lim_{t \to \infty} e(t) = 0$.

Definition 2 (Error feedback regulator problem) Consider the system (1), driven by the exosystem (2) and interconnected with the controller (4). The error feedback regulator problem is the problem of determining the matrices F, G, and H of the controller such that:

(S) The system

$$
\begin{aligned}
\sigma x &= Ax + BH\chi, \\
\sigma \chi &= F\chi + GCx,
\end{aligned}
$$

is asymptotically stable.

(R) All trajectories of the system

$$
\begin{aligned}
\sigma d &= Sd, \\
\sigma x &= Ax + BH\chi + Pd, \\
\sigma \chi &= F\chi + G(Cx + Qd), \\
e &= Cx + Qd,
\end{aligned} \quad (6)
$$

are such that $\lim_{t \to \infty} e(t) = 0$.

The Full Information Regulator Problem

Consider the full information regulator problem and assume the following.

Assumption 1 The matrix S of the exosystem has all eigenvalues with nonnegative real part, in the case of continuous-time systems, or with modulo not smaller than one, in the case of discrete-time systems.

Assumption 2 The system (1) with $d = 0$ is reachable.

Assumption 1 implies that there are no initial conditions $d(0)$ such that the signal $d(t)$ converges (asymptotically) to zero. This assumption is not restrictive. In fact, disturbances converging to zero do not have any effect on the asymptotic behavior of the system, and references which converge to zero can be tracked simply by driving the state of the system to zero, i.e., by stabilizing the system. Assumption 2 implies that it is possible to arbitrarily assign the eigenvalues of the matrix $A + BK$ by a proper selection of K. Note that, in practice, this assumption can be replaced by the weaker assumption that the system (1) with $d = 0$ is stabilizable.

We now present a preliminary result which is instrumental to derive a solution to the full information regulator problem.

Lemma 1 *Consider the full information regulator problem. Suppose Assumption 1 holds. Suppose, in addition, that there exist matrices K and L such that condition (S) holds.*

Then condition (R) holds if and only if there exists a matrix $\Pi \in \mathbb{R}^{n \times r}$ such that the equations

$$
\begin{aligned}
\Pi S &= (A + BK)\Pi + (P + BL), \\
0 &= C\Pi + Q,
\end{aligned} \quad (7)
$$

hold.

Proof Consider the system (5) and the coordinates transformation

$$
\begin{aligned}
\hat{d} &= d, \\
\hat{x} &= x - \Pi d,
\end{aligned}
$$

where Π is the solution of the equation

$$
\Pi S = (A + BK)\Pi + (P + BL).
$$

This equation is a so-called Sylvester equation. The Sylvester equation is a (matrix) equation of the form

$$
A_1 X = XA_2 + A_3,
$$

in the unknown X. This equation has a unique solution, for any A_3, if and only if the matrices A_1 and A_2 do not have common eigenvalues. Note that, by condition (S) and Assumption 1, there is a unique matrix Π which solves this equation. In the new coordinates $\hat{x}$ and $\hat{d}$, the system is described by the equations

$$\sigma \hat{d} = S\hat{d},$$
$$\sigma \hat{x} = (A + BK)\hat{x},$$
$$e = C\hat{x} + (C\Pi + Q)\hat{d}.$$

By condition (S) $\lim_{t \to \infty} \hat{x}(t) = 0$, hence condition (R) holds, by Assumption 1, if and only if $C\Pi + Q = 0$. In summary, under the stated assumptions, condition (R) holds if and only if there exists a matrix Π such that Eqs. (7) hold.

We are now ready to state and prove the result which provides conditions for the solvability of the full information regulator problem.

Theorem 1 *Consider the full information regulator problem. Suppose Assumptions 1 and 2 hold. There exists a full information control law described by Eq. (3) which solves the full information regulator problem if and only if there exist two matrices Π and Γ such that the equations*

$$\Pi S = A\Pi + B\Gamma + P,$$
$$0 = C\Pi + Q, \tag{8}$$

hold.

Proof (Necessity) Suppose there exist two matrices K and L such that conditions (S) and (R) of the full information regulator problem hold. Then, by Lemma 1, there exists a matrix Π such that Eqs. (7) hold. As a result, the matrices Π and $\Gamma = K\Pi + L$ are such that Eqs. (8) hold.

(Sufficiency) The proof of the sufficiency is constructive. Suppose there are two matrices Π and Γ such that Eqs. (8) hold. The full information regulator problem is solved selecting K and L as follows. The matrix K is any matrix such that the system $\sigma x = (A + BK)x$ is asymptotically stable. By Assumption 2,

such a matrix K does exist. The matrix L is selected as $L = \Gamma - K\Pi$. This selection is such that condition (S) of the full information regulator problem holds; hence, to complete the proof, we have only to show that, with K and L as selected above, Eqs. (7) hold. This is trivially the case. In fact, replacing L in (7) yields Eqs. (8), which hold by assumption. As a result, also condition (R) of the full information regulator problem holds, and this completes the proof.

The proof of Theorem 1 implies that a controller which solves the full information regulator problem is described by the equation

$$u = Kx + (\Gamma - K\Pi)d,$$

with K such that a stability condition holds, and Π and Γ such that Eqs. (8) hold. By Assumption 2, the stability condition can be always satisfied. As a result, the solution of the full information regulator problem relies upon the existence of a solution of Eqs. (8).

The FBI Equations

Equations (8), known as the Francis-Byrnes-Isidori (FBI) equations, are linear equations in the unknowns Π and Γ, for which the following statement holds.

Lemma 2 *Equations (8), in the unknowns Π and Γ, are solvable for any P and Q if and only if*

$$\text{rank} \begin{bmatrix} sI - A & B \\ C & 0 \end{bmatrix} = n + p, \tag{9}$$

for all s which are eigenvalues of the matrix S.

For single-input, single-output systems (i.e., $m = p = 1$), the condition expressed by Lemma 2 has a very simple interpretation. In fact, the complex numbers s such that

$$\text{rank} \begin{bmatrix} sI - A & B \\ C & 0 \end{bmatrix} < n + 1$$

are the zeros of the system

$$\sigma x = Ax + Bu,$$
$$y = Cx,$$

that is the roots of the numerator polynomial of the transfer function $W(s) = C(sI - A)^{-1}B$, i.e., the zeros of $W(s)$. This implies that, for single-input, single-output systems, the full information regulator problem is solvable if and only if the eigenvalues of the exosystem are not zeros of the transfer function of the system (1) with input u, output e, and $d = 0$.

The Error Feedback Regulator Problem

To provide a solution to the error feedback regulator problem, we need to introduce a new assumption.

Assumption 3 The system

$$\begin{bmatrix} \sigma x \\ \sigma d \end{bmatrix} = \begin{bmatrix} A & P \\ 0 & S \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix},$$
$$e = \begin{bmatrix} C & Q \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix} \tag{10}$$

is observable.

Note that Assumption 3 implies observability of the system

$$\sigma x = Ax,$$
$$y = Cx. \tag{11}$$

To show this property, note that observability of the system (10) implies that

$$\text{rank} \begin{bmatrix} C & Q \\ CA & \vdots \\ \vdots & \vdots \\ CA^{n+r-1} & \vdots \end{bmatrix} = n + r.$$

This, in turn, implies

$$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n+r-1} \end{bmatrix} = n$$

and, by Cayley-Hamilton Theorem,

$$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n,$$

which implies observability of system (11). Similarly to what discussed in the case of Assumption 2, Assumption 3 can be replaced by the weaker assumption that the system (10) is detectable. We are now ready to state and prove the result which provides conditions for the solvability of the error feedback regulator problem.

Theorem 2 *Consider the error feedback regulator problem. Suppose Assumptions 1–3 hold. There exists an error feedback control law described by Eq. (4) which solves the full information regulator problem if and only if there exist two matrices Π and Γ such that the equations*

$$\Pi S = A\Pi + B\Gamma + P,$$
$$0 = C\Pi + Q, \tag{12}$$

hold.

Remark Theorem 2 can be alternatively stated as follows. Consider the error feedback regulator problem. Suppose Assumptions 1–3 hold. Then the error feedback regulator problem is solvable if and only if the full information regulator problem is solvable.

Proof (Necessity) The proof of the necessity is similar to the proof of the necessity of Theorem 1, hence omitted.

(Sufficiency) The proof of the sufficiency is constructive. Suppose there are two matrices Π and Γ such that Eqs. (12) hold. Then, by Theorem 1, the full information control law $u = Kx + (\Gamma - K\Pi)d$, with K such that the system $\sigma x = (A + BK)x$ is asymptotically stable,

solves the full information regulator problem. This control law is not implementable, because we only measure e. However, by Assumption 3, it is possible to build asymptotic estimates ξ and δ of x and d; hence, implement the control law

$$u = K\xi + (\Gamma - K\Pi)\delta. \tag{13}$$

To this end, consider an observer described by the equation

$$\begin{bmatrix} \sigma\xi \\ \sigma\delta \end{bmatrix} = \begin{bmatrix} A & P \\ 0 & S \end{bmatrix} \begin{bmatrix} \xi \\ \delta \end{bmatrix}$$
$$+ \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \left(\begin{bmatrix} C & Q \end{bmatrix} \begin{bmatrix} \xi \\ \delta \end{bmatrix} - e \right)$$
$$+ \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} K & \Gamma - K\Pi \end{bmatrix} \begin{bmatrix} \xi \\ \delta \end{bmatrix}.$$

The estimation errors $e_x = x - \xi$ and $e_d = d - \delta$ are such that

$$\begin{bmatrix} \sigma e_x \\ \sigma e_d \end{bmatrix} = \left(\begin{bmatrix} A & P \\ 0 & S \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \begin{bmatrix} C & Q \end{bmatrix} \right)$$
$$\begin{bmatrix} e_x \\ e_d \end{bmatrix}; \tag{14}$$

hence, by Assumption 3, there exist G_1 and G_2 that assign the eigenvalues of this error system. Note now that the control law (13) can be rewritten as $u = Kx + (\Gamma - K\Pi)d - (Ke_x + (\Gamma - K\Pi)e_d)$; hence, the control law is composed of the full information control law, which solves the regulator problem, and of an additive disturbance, which decays exponentially to zero. Such a disturbance does not affect the regulation requirement, provided the closed-loop system is asymptotically stable. Therefore, to complete the proof, we need to show that condition (S) holds. In the coordinates x, e_x, and e_d, the closed-loop system, with $d = 0$, is described by the equations

$$\begin{bmatrix} \sigma x \\ \sigma e_x \\ \sigma e_d \end{bmatrix} = \begin{bmatrix} A + BK & -BK & -B(\Gamma - K\Pi) \\ 0 & A + G_1C & P + G_1Q \\ 0 & G_2C & S + G_2Q \end{bmatrix}$$
$$\begin{bmatrix} x \\ e_x \\ e_d \end{bmatrix}. \tag{15}$$

Recall that the matrices G_1 and G_2 have been selected to render system (14) asymptotically stable and that K is such that the system $\sigma x = (A + BK)x$ is asymptotically stable. As a result, system (15) is asymptotically stable.

The Internal Model Principle

The proof of Theorem 2 implies that a controller which solves the error feedback regulator problem is described by equations of the form (4) with

$$\chi = \begin{bmatrix} \xi \\ \delta \end{bmatrix},$$

$$F = \begin{bmatrix} A + G_1C + BK & P + G_1Q + B(\Gamma - K\Pi) \\ G_2C & S + G_2Q \end{bmatrix}, \tag{16}$$

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \qquad H = \begin{bmatrix} K & \Gamma - K\Pi \end{bmatrix},$$

K, G_1, and G_2 such that a stability condition holds and Π and Γ such that Eqs. (12) hold. This controller, and in particular the matrix F, possesses a very interesting property.

Proposition 1 (Internal model property) *The matrix F in Eq. (16) is such that*

$$F\Sigma = \Sigma S,$$

for some matrix Σ of rank r. In particular, any eigenvalue of S is also an eigenvalue of F.

Proof Let

$$\Sigma = \begin{bmatrix} \Pi \\ I \end{bmatrix}$$

and note that $\text{rank}\,\Sigma = r$, by construction, and that

$$F\Sigma = \begin{bmatrix} A\Pi + G_1 C\Pi + BK\Pi + P + G_1 Q \\ + B(\Gamma - K\Pi) - G_2 C\Pi + S - G_2 Q \end{bmatrix}$$

$$= \begin{bmatrix} (A\Pi + B\Gamma + P) + G_1(C\Pi + Q) \\ S - G_2(C\Pi + Q) \end{bmatrix}$$

$$= \begin{bmatrix} \Pi S \\ S \end{bmatrix} = \Sigma S,$$

hence the first claim. To prove the second claim, let λ be an eigenvalue of S and v the corresponding eigenvector. Then $Sv = \lambda v$; hence,

$$F\Sigma v = \Sigma Sv = \lambda \Sigma v,$$

which shows that λ is an eigenvalue of F with eigenvector Σv, and this proves the second claim.

It is possible to prove that the property highlighted in Proposition 1 is shared by all error feedback control laws which solve the considered regulation problem. This property, which is often referred to as the internal model principle, can be interpreted as follows. The control law solving the regulator problem has to *contain* a copy of the exosystem, i.e., it has to be able to generate, when $e = 0$, a copy of the exogenous signal.

Summary and Future Directions

The problem of tracking and regulation for linear systems in the presence of references and/or disturbances generated by a linear signal generator has been solved. It has been shown that the problem is solvable provided a system of linear matrix equations admits a solution. The tracking and regulation problem can be studied and solved for more general classes of systems, including nonlinear systems, distributed parameter systems, and hybrid systems, exploiting the same ideas presented in this article.

Cross-References

Bibliography

Classical references on the tracking and regulation problem for linear systems are given below.

Gruyitch, LT (2013) Tracking control of linear systems. CRC, Boca Raton

Wonham, WM (1985) Linear multivariable control: a geometric approach, 3rd edn. Springer, New York

Tracking Model Predictive Control

Daniel Limon and Teodoro Alamo
Departamento de Ingeniería de Sistemas y Automática, Escuela Superior de Ingeniería, Universidad de Sevilla, Sevilla, Spain

Abstract

The main objective of tracking model predictive control is to steer the tracking error, that is, the difference between the reference and the output, to zero while the constraints are satisfied. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

The most extended case is to assume that the reference will remain constant along the prediction horizon. Tracking predictive schemes for constant references are typically based on a two-layer control structure in which, provided the value of the reference, first, an appropriate set point is computed and then a nominal MPC

T

is designed to steer the system to this target. Under certain assumptions, closed-loop stability can be guaranteed if the initial state is inside the feasibility region of the MPC. However, if the value of the reference is changed, then there is no guarantee that feasibility and stability properties of the resulting control law hold. Specialized predictive controllers have been designed to deal with this problem. Particularly interesting is the so-called MPC for tracking, which ensures recursive feasibility and asymptotic stability of the set point when the value of the reference is changed.

The presence of exogenous disturbances or model mismatches may lead to the controlled system to exhibit offset error. Offset-free control in the presence of unmeasured disturbances can be addressed by using disturbance models and disturbance estimators together with the tracking predictive controller.

Keywords

Loss of feasibility; MPC for tracking; Offset-free control; Set-point tracking

Introduction

The problem of designing and stabilizing model predictive control (MPC) schemes to regulate a system to the origin has been widely studied, and there are well-known solutions for varied cases including linear, nonlinear, and uncertain systems, among others (Rawlings and Mayne 2009).

The objective of tracking MPC is to ensure a tracking error, which is the difference between a reference or desired output r and the actual output y, tends to zero.

The most common tracking problem is when the reference r is constant. In this case, the controller is required to steer the state x of the plant and the control input u applied to the plant to a set point (x_r, u_r) where the tracking error y_r is zero and the plant is in equilibrium (at rest); the state x_r is called a target. It is also necessary to ensure that x_r is asymptotically stable for the controlled

system, i.e., that the state x converges to x_r and that, near x_r, small changes in x cause small changes in the subsequent trajectory. A relatively straightforward solution for this problem exists.

Set-point tracking is a relevant control problem in the process industry in which the plant is typically designed to operate at an equilibrium point that maximizes the profit of the plant. In this case, the optimal set point is calculated online by a real-time optimizer (RTO) according to an economic criteria. The set points remain constant for a long period of time, until the RTO, which is executed at a very low frequency, calculates a different set point. The steady-state target associated to the given set point must be calculated and provided to the MPC to track this target.

The tracking problem is considerably more difficult when the reference r varies in a way not known a priori because MPC is naturally suited to deterministic control problems. Uncertainty requires the "invention" of special techniques so that a variety of solutions have been proposed in the literature to deal with a varying reference (Bemporad et al. 1997; Chisci and Zappa 2003; Limon et al. 2008; Maeder and Morari 2010; Pannocchia and Rawlings 2003; Rossiter et al. 1996).

Another tracking problem arises when there exists a mismatch between the model used for prediction in the optimal control problem and the real plant. If the reference is constant and the model mismatch is sufficiently small not to cause loss of asymptotic stability, the state and control will converge to values at which the predicted tracking error, but not the actual tracking error, is zero. The difference between the predicted and actual values of the output y is known as the offset; offset-free tracking when the reference is constant may be achieved by incorporation of a suitable observer to estimate the offset.

Notation

The set $\mathbb{I}_M$ denotes the set of integers $\{0, 1, \cdots, M\}$. I_n denotes the identity matrix in $\mathbb{R}^{n \times n}$. $\mathbf{z}$ denotes a signal (or time sequence) $\mathbf{z} = \{z(0), z(1), \cdots\}$, whose cardinality is inferred from the context. A signal that depends on a parameter θ is denoted as $\mathbf{z}(\theta)$ and $z(i; \theta)$

denotes its ith element. A closed polyhedron $\mathcal{X} \subset \mathbb{R}^n$ is a set that results of the intersection of a finite number of hyperplanes as follows: $\mathcal{X} = \bigcap_i \{x : F_i x \leq f_i\}$, where $F_i \in \mathbb{R}^{1 \times n}$ and $f_i \in \mathbb{R}$.

Problem Statement

In this article, for the sake of simplicity, we consider that the system to be controlled can be modeled as a linear time-invariant system described by a discrete-time state-space linear model:

$$x(k + 1) = Ax(k) + Bu(k) \qquad (1a)$$

$$y(k) = Cx(k) \qquad (1b)$$

where $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}^m$, and $y(k) \in \mathbb{R}^p$ are the state, the manipulable inputs, and the outputs of the system at time step k, respectively. This model will be used to calculate the predictions in the predictive controller.

The evolution of the plant must be such that the constraint

$$(x(k), u(k)) \in \mathcal{Z} \qquad (2)$$

is satisfied for all $k \geq 0$. The set $\mathcal{Z}$ is a closed polyhedron. Without loss of generality, we assume that $(0, 0) \in \mathcal{Z}$.

The main objective of tracking model predictive control is to steer the system output to the reference, that is, steer the tracking error $y - r$ to zero, while the constraints are satisfied. In order to predict the expected evolution of the tracking error, some assumptions on the future values of the reference must be considered. Since the reference may differ from expected, the tracking problem is inherently uncertain.

Thus, assuming that the reference signal is known a priori, $\mathbf{r} = \{r(0), r(1), \cdots\}$, the tracking model predictive control law $\kappa(x(k), \mathbf{r})$ must be designed to ensure that the resulting controlled system

$$x(k + 1) = Ax(k) + B\kappa(x(k), \mathbf{r})$$

$$y(k) = Cx(k)$$

satisfies the constraints, i.e., $(x(k), u(k)) \in \mathcal{Z}$ for all $k \geq 0$ is stable and, if it is possible, the controlled output converges to the reference, that is,

$$\lim_{k \to \infty} \|y(k) - r(k)\| = 0.$$

It is assumed that the system is stabilizable and that the outputs are linearly independent. It is also considered that the state is measured and available at each sample.

Tracking MPC for a Constant Reference

The most simple tracking problem is to consider that the reference signal is a constant signal in the future equal to the actual value of the reference, i.e., $r(k) = r$. This control problem is very common in the process industry, for instance, where processes are typically designed to operate at certain equilibrium point.

Determining the Set Point

Corresponding to each value r of the reference is a set point (x_r, u_r) that is ideally an equilibrium point of the prediction model, i.e., it satisfies

$$x_r = Ax_r + Bu_r. \qquad (3)$$

The set point (x_r, u_r) is also required to satisfy

$$y_r = Cx_r = r \qquad (4)$$

and

$$(x_r, u_r) \in \mathcal{Z}$$

so that the tracking error $y - r$ is zero and the constraint (2) is satisfied at the set point. Because the set point is an equilibrium point, the tracking error remains zero once the set point is reached.

Conditions for the existence of a set point possessing the above properties are given in Rawlings and Mayne (2009, Lemma 1.14).

In practice, the condition $(x_r, u_r) \in \mathcal{Z}$ is replaced by $(x_r, u_r) \in \mathcal{Z}_s \subset \text{interior}\{\mathcal{Z}\}$ in

order to ensure that the constraint $(x_r, u_r) \in \mathcal{Z}$ is not active at the set point, and the tracking error requirement is slightly relaxed so that the set point is determined by solving

$$(x_r, u_r) = \arg \min_{(x_s, u_s) \in \mathcal{Z}_s} \ell_t(x_s, u_s, r) \quad (5)$$

where ℓ_t is a convex function, typically a quadratic function as follows:

$$\ell_t(x_s, u_s, r) = \|Cx_s - r\|_{Q_s}^2 + \|u_s\|_{R_s}^2$$

This problem is referred to as steady-state target optimization problem (Rao and Rawlings 1999).

Model Predictive Controller Design

If the reference to be tracked is a constant, i.e., $r(k) = r$ for all k, then the control objective is to stabilize the system and steer the initial state $x(0)$ to the set-point state x_r. As is usual in model predictive control, a finite horizon optimization problem that depends on the current state x and the constant reference r is solved yielding a control sequence $\mathbf{u}^o(x, r) = \{u^o(0; x, r), u^o(1; x, r), \cdots, u^o(N-1; x, r)\}$ and the associated state trajectory $\mathbf{x}^o(x, r) = \{x^o(0; x, r) = x, x^o(1; x, r), \cdots, x^o(N; x, r)\}$, where N is the prediction horizon. The first element of this sequence, namely, $u^o(0; x, r)$, is applied to the system.

Because the reference is constant, the appropriate optimal control problem $P_N(x, r)$ is a slight variation of that discussed in the article ▶ Nominal Model-Predictive Control and is defined by

$$\min_{\mathbf{u}} \sum_{j=0}^{N-1} \ell(x(j), u(j), r) + V_f(x(N), r)$$

$$s.t. \ x(0) = x, \quad (6a)$$

$$x(j+1) = Ax(j) + Bu(j),$$
$$j \in \mathbb{I}_{N-1} \quad (6b)$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \quad (6c)$$

$$x(N) \in X_f(r) \quad (6d)$$

The stage cost function $\ell(\cdot)$ is a measure of the predicted tracking error set point, that is, $\ell(x_r, u_r, r) = 0$ and $\ell(x, u, r) \geq \alpha_1(\|x - x_r\|)$. The terminal cost function $V_f(\cdot)$ is such that

$$\alpha_2(\|x - x_r\|) \leq V_f(x_r, r) \leq \alpha_3(\|x - x_r\|).$$

Functions α_i are $\mathcal{K}_\infty$ functions (see the article ▶ Nominal Model-Predictive Control). The set of states where this optimization problem is feasible is denoted as $X_N(r)$.

The solution of the optimal control problem $P_N(x, r)$ yields the receding horizon control law

$$\kappa_N(x, r) = u^o(0; x, r)$$

and the system under model predictive control satisfies

$$x(k+1) = Ax(k) + B\kappa_N(x(k), r) \quad (7)$$

Because the horizon N is finite, x_r is not necessarily asymptotically stable for this system, but asymptotic stability can be ensured if the terminal cost function $V_f(\cdot)$ and the terminal region $X_f(r)$ are chosen appropriately.

The functions $\ell(\cdot)$, $V_f(\cdot)$ and the set $X_f(r)$ must satisfy the following condition.

Stability conditions for nominal MPC: For all $x \in X_f(r)$, there exists a control input u such that $(x, u) \in \mathcal{Z}$ and the successor state $x^+ = Ax + Bu$ are contained in $X_f(r)$ and

$$V_f(x^+, r) - V_f(x, r) \leq -\ell(x, u, r).$$

These conditions are trivially satisfied taking $X_f(r) = x_r$ and $V_f(x, r) = 0$.

Under these assumptions, the optimization problem is recursively feasible, i.e., if $P_N(x(0), r)$ is feasible, then all subsequent problems $P_N(x(i), r)$ are also feasible. Besides, the optimal cost function is a Lyapunov function of the system (7). Then, the set point (x_r, u_r) is an asymptotically stable equilibrium point of the system (7) and the domain of attraction is $X_N(r)$.

Tracking MPC for a Changing Reference

The previous predictive controller is inherently deterministic, since it is assumed that the reference is known and this will remain constant in the future. However, in a realistic scenario, the reference may be changed without a predefined deterministic law or even randomly. In this section, a tracking predictive controller, for the case when the reference is constant or varying but ultimately constant, is presented.

Feasibility and Stability Issues

If the reference r is constant, tracking MPC ensures asymptotic stability of the target state x_r and convergence to zero of the tracking error $y - r$. However, if the reference r varies, recursive feasibility (i.e., feasibility of $P_N(x(k), r(k))$ at each time instant k) and asymptotic stability may be compromised. For each value of r, the feasibility region $X_N(r)$ is the set of states for which $P_N(x, r)$ has a solution; it is also the domain of attraction for the closed-loop system (7). If r changes value from r_1 to r_2, the terminal constraint set $X_f(r_2)$ and the terminal cost function $V_f(\cdot, r_2)$ have to be computed. The current state

x, which lies in $X_N(r_1)$, does not necessarily lie in $X_N(r_2)$ so that $\kappa_N(\cdot, r_2)$ is undefined and the model predictive controller fails.

This phenomenon is illustrated for the double integrator system where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \ B = \begin{bmatrix} 0 & 0.5 \\ 1 & 0.5 \end{bmatrix}, \ C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and the set of constraints is given by

$$\mathcal{Z} = \{(x, u) : \|x\|_\infty \leq 5, \ \|u\|_\infty \leq 0.3\}$$

The initial state is $x(0) = (2.91, -1.83)$ and the initial value of the reference is $r_1 = -2$. The corresponding set point is (x_{r_1}, u_{r_1}) where $x_{r_1} = (-2, 0)$ and $u_{r_1} = (0, 0)$. If the reference changes from r_1 to r_2, the new set point is (x_{r_2}, u_{r_2}) where $x_{r_2} = (4.5, 0)$ and $u_{r_2} = (0, 0)$. The horizon is chosen to be $N = 3$ and the domains of attraction for the two values of r are, respectively, $X_3(r_1)$ and $X_3(r_2)$. These two domains, $X_3(r_1)$ and $X_3(r_2)$, are disjoint. While $r = r_1$, the state trajectory commencing at $x(0) \in X_3(r_1)$ remains in $X_3(r_1)$. If r subsequently changes its value to r_2 at time t_1, the model predictive controller

Tracking Model Predictive Control, Fig. 1 Example of the double integrator: terminal regions ($X_f(r_1)$ and $X_f(r_2)$) and domains of attraction of MPC ($X_3(r_1)$, $X_3(r_2)$, and $X_{10}(r_2)$)
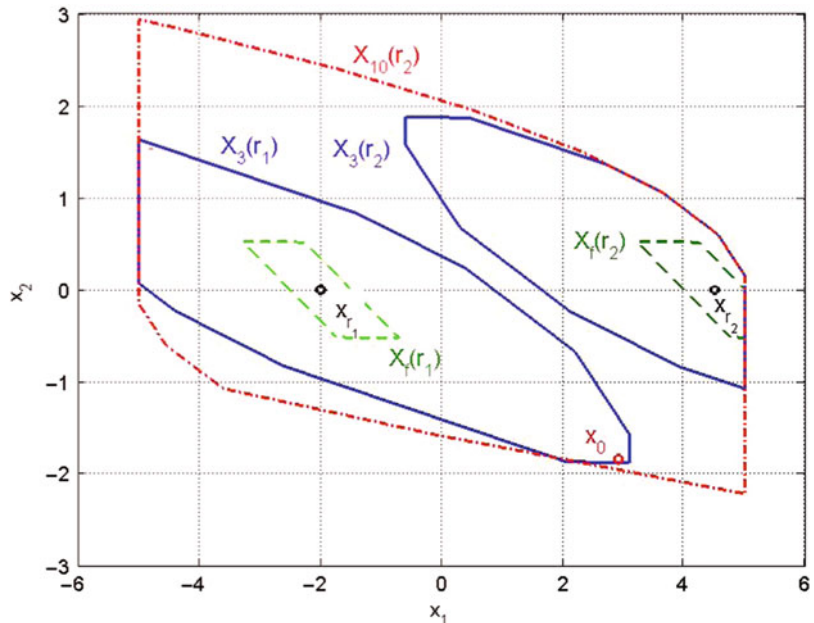
fails since $x(t_1)$ does not lie in $X_3(r_2)$. This is illustrated in Fig. 1.

These feasibility and stability issues can be overcome if the predictive controller is redesigned for the new set point. This would require the calculation of a new terminal set and a prediction horizon each time the set point changes. For instance, in the example of Fig. 1, if the terminal constraint is recalculated for r_2 and the prediction horizon is chosen as $N = 10$, then the MPC controller steers the system to the reference r_2 since $x(0) \in X_{10}(r_2)$. This recalculation can be done off-line if the set-point changes are a priori known (Findeisen et al. 2000; Wan and Kothare 2003). Other methods to avoid this issue are designing a predictive controller to provide a certain degree of robustness to set-point variations (Pannocchia 2004; Pannocchia and Kerrigan 2005) and a predictive control law with a mode to recover recursive feasibility (Chisci and Zappa 2003; Rossiter et al. 1996) or using specialized predictive control laws (Magni and Scattolini 2005; Magni et al. 2001). Another solution to this case is to use a reference governor and a predictive controller (Bemporad et al. 1997; Olaru and Dumur 2005).

Stabilizing MPC for Tracking

The idea behind the reference governor is to introduce an artificial reference r^a that is manipulated to ensure that the current state is in the domain of attraction $X_N(r^a)$ while tends to the actual reference r if r remains constant or tends to a constant. In Limon et al. (2008), this idea is used to formulate the MPC for tracking. The artificial reference r^a is an extra decision variable in the optimal control problem to avoid the loss of feasibility issue. In order to enforce the convergence to the actual reference r, a term that penalizes the deviation between the artificial reference r^a and the actual reference r, $\ell_o(r^a, r)$ is added. This function is assumed to be convex in r^a. A suitable choice of this term is the cost function of the steady-state target calculator (5), i.e., $\ell_o(r^a, r) = \ell_t(x_{r^a}, u_{r^a}, r)$, where (x_{r^a}, u_{r^a}) is the artificial set point associated to the artificial reference r^a.

The optimal model predictive control problem $P_N^t(x, r)$ for tracking is given by

$$\min_{\mathbf{u}, r^a} \sum_{j=0}^{N-1} \ell(x(j), u(j), r^a) + V_f(x(N), r^a)$$
$$+ \ell_o(r^a, r) s.t. x(0) = x, \tag{8a}$$

$$x(j + 1) = Ax(j) + Bu(j),$$
$$j \in \mathbb{I}_{N-1} \tag{8b}$$

$$(x(j), u(j)) \in \mathcal{Z}, \quad j \in \mathbb{I}_{N-1} \tag{8c}$$

$$r^a \in \mathcal{R} \tag{8d}$$

$$(x(N), r^a) \in \Gamma \tag{8e}$$

where $\mathcal{R} = \{r : (x_r, u_r) \in \mathcal{Z}_s, Ax_r + Bu_r = x_r, Cx_r = r\}$.

Condition (3) is an extended terminal constraint of both the terminal state $x(N)$ and the artificial reference r^a. The feasibility region of this optimization problem X_N^t is the set of states that can be steered to any reference of the set $\mathcal{R}$ in N steps, that is,

$$X_N^t = \bigcup_{r^a \in \mathcal{R}} X_N(r^a)$$

The terminal cost function $V_f(\cdot)$ and the terminal constraint set, Γ, must satisfy appropriately modified stability conditions in order to ensure recursive feasibility and asymptotic stability of (x_r, u_r). The stability conditions are the following.

Stability conditions for tracking MPC: For all $(x, r^a) \in \Gamma$, there exists a u satisfying:
(i) $(x, u) \in \mathcal{Z}$
(ii) the successor state $x^+ = Ax + Bu$ such that $(x^+, r^a) \in \Gamma$ and
$V_f(x^+, r^a) - V_f(x, r^a) \leq -\ell(x, u, r^a)$.

As shown in Limon et al. (2008), if the terminal control law is chosen as $u = K(x - x_{r^a}) + u_{r^a}$ with K such that the eigenvalues of $A + BK$ are in the unitary disk, then the terminal set Γ can be calculated using standard algorithms to compute positively invariant sets for constrained linear

systems and it is a polyhedron. A simple choice of the terminal cost and constraint satisfying these assumptions is $V_f(\cdot) = 0$ and $\Gamma = \{(x, r^a) : x = x_{r^a}\}$.

Theorem 1 *If the stability conditions for tracking MPC hold, then predictive control law derived from the optimal control problem $P_N^t(x, r)$ is such that:*

1. *For all feasible initial state, i.e., $x(0) \in X_N^t$, and for all $r \in \mathbb{R}^p$, the optimization problem is recursively feasible, that is, if $P_N^t(x(0), r)$ is feasible, then all the subsequent problems $P_N^t(x(i), r)$ are also feasible.*
2. *If r is admissible, i.e., $r \in \mathcal{R}$, then the set point (x_r, u_r) is an asymptotically stable equilibrium point of the closed-loop system and the domain of attraction is X_N^t.*
3. *If r is not admissible, that is, $r \notin \mathcal{R}$, then the set point (x_{r*}, u_{r*}) such that*

$$r^* = \arg \min_{r^a \in \mathcal{R}} \ell_o(r^a, r)$$

   is asymptotically stable and the domain of attraction is X_N^t.
4. *The domain of attraction X_N^t is larger than the domain of the nominal MPC for any reference $r \in \mathcal{R}$, that is, $X_N(r) \subseteq X_N^t$, and contains all the equilibrium points contained in $\mathcal{Z}_s$.*
5. *If the reference $r(k)$ is not constant and converges to a steady value r, the optimization problem is recursively feasible and the set point (x_r, u_r) is an asymptotically stable equilibrium point for all $x(0) \in X_N^t$.*

In Fig. 2a the aforementioned properties are illustrated for the example of the double integrator. The MPC for tracking has been designed with the same prediction horizon $N = 3$ and the same terminal control law and the terminal cost function that in the previous tracking MPC case. The initial state is also the same and the reference signal is $r(k) = r_2$ for $k \leq 30$ and $r(k) = r_1$ for $k > 30$. Notice that the tracking MPC cannot be used to do this without redesign. In Fig. 2a, it can be seen that the domain of attraction of the MPC for tracking X_3^t is larger than the domain provided by the standard tracking MPC $X_3(r_1)$ or $X_3(r_2)$.

This figure also shows the state portrait of the closed-loop trajectory. In Fig. 2b the trajectories of the reference signal **r**, the controlled output **y**, and the artificial target output $y_{r^a} = Cx_{r^a}$ are depicted. Notice the role of the artificial target: y_{r^a} differs from the reference in order to guarantee recursive feasibility and finally converges to the reference r to enforce asymptotic stability.

Offset-Free Tracking

In practice there may exist mismatches between the prediction model and the dynamics of the real plant to be controlled, due, for instance, to un-modeled nonlinearities or unmeasured disturbances. This would require to design the predictive controller to be robust to this uncertain effects. Assuming that the predictive controller based on nominal predictions is robustly stable and considering that the controlled system converges to a steady state, there may exist a steady error between set point and the output.

This offset can be canceled taking into account a prediction model corrected by a disturbance model (Pannocchia and Rawlings 2003). To achieve offset-free control, the disturbance is assumed to be an integrating disturbance as follows:

$$x(k + 1) = Ax(k) + Bu(k) + B_d d(k) \quad \text{(9a)}$$

$$d(k + 1) = d(k) \quad \text{(9b)}$$

$$y(k) = Cx(k) + D_d d(k) \quad \text{(9c)}$$

Matrices B_d and D_d define the disturbance model and these are chosen to guarantee offset-free control. They are typically chosen as $B_d = 0$ and $D_d = I_p$.

The disturbance signal $d(k)$ is estimated using an observer based on the disturbance model. The disturbance model and the estimator gains can be calculated separately, but this may lead to a poor closed-loop performance. A joint design procedure has been proposed in Pannocchia and Bemporad (2007).
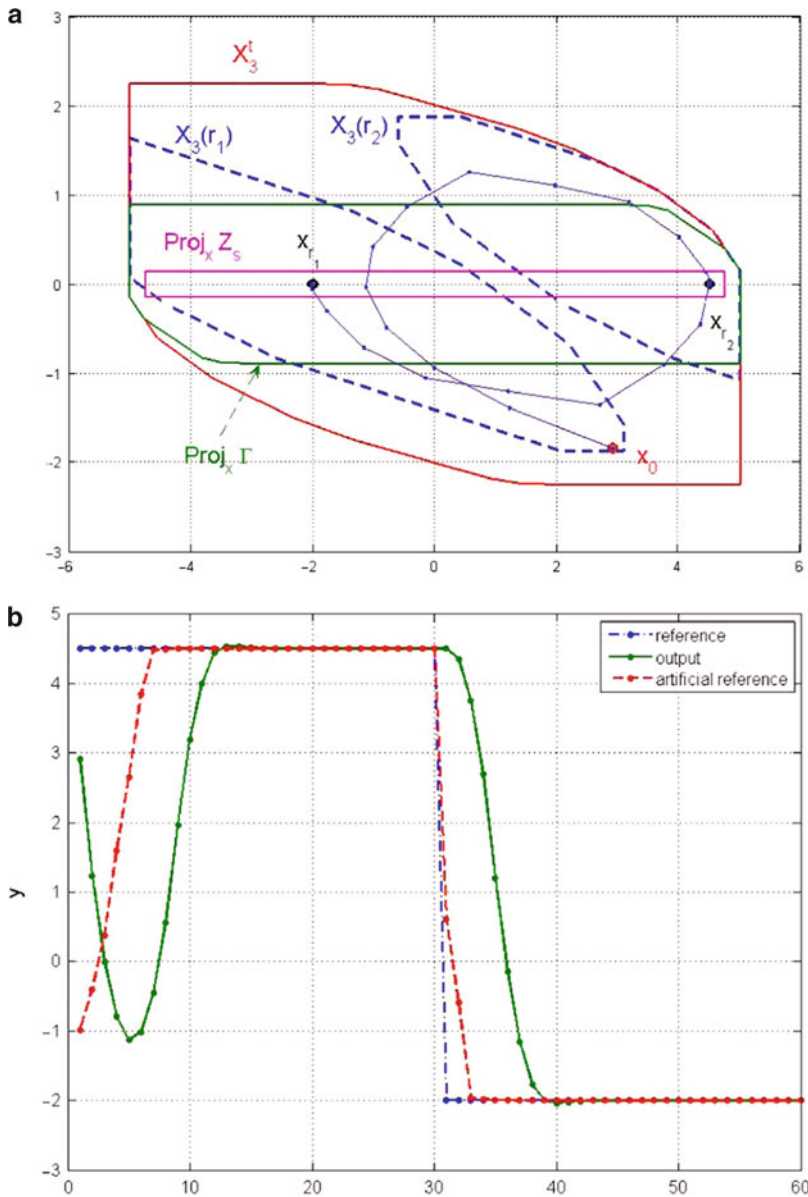
Tracking Model Predictive Control, Fig. 2 The double integrator controlled by the MPC for tracking. (**a**) Comparison of the domains of attraction of the tracking MPC $X_3(r_1)$ and $X_3(r_2)$ vs. the domain of attraction of the MPC for tracking X_3^t. (**b**) Trajectories of the reference, the controlled output, and the artificial reference r^a

Once the estimated disturbance $\hat{d}$ is available, the corrected prediction model (9) must be used to calculate the MPC target in the steady-state target optimization problem (5) (x_r, u_r) and to calculate the predictions in the optimization problem $P_N(x, x_r, u_r, \hat{d})$.

Future Directions

Tracking model predictive control is an inherently uncertain control problem due to the unexpected changes in the reference. Constant reference tracking has been widely studied and there exist a number of nice solutions.

The case of trajectory tracking is not as mature as the set-point tracking case. If the reference signal is known a priori, this can be used to calculate the predicted cost. This control problem can be solved by using a two-layer structure: a trajectory planning on top of a predictive control law that steers the system to the trajectory target. Asymptotic stability to the trajectory target can be proved using terminal equality constraint resorting on the regulation problem. Another interesting line is to assume that the reference is the output of a certain dynamic system. For different families of trajectories, such as ramps or sinusoidal signals, Maeder and Morari (2010) has proposed a reference tracking MPC based on extended disturbance models.

The problem of tracking MPC in case of unknown (or changing) reference signals can be considered an open problem that deserves more research efforts.

Another interesting control problem is the tracking of unreachable (equilibrium point as well as trajectory) targets. Recently this problem has been posed as an economic model predictive control problem (Rawlings and Mayne 2009). Therefore, the stabilizing design of economic MPC presented in Angeli et al. (2012) can be extended to the case of tracking unreachable targets.

Cross-References

▶ Economic Model Predictive Control
▶ Nominal Model-Predictive Control
▶ Regulation and Tracking of Nonlinear Systems
▶ Tracking and Regulation in Linear Systems

Recommended Reading

The book Camacho and Bordons (2004) covers the classic approach to the tracking MPC. In Rawlings and Mayne (2009), the authors deal with the tracking MPC in a very general and clear way and survey existing results on stability, target calculation, and offset-free control for linear and nonlinear models. In Muske (1997), the

reachability of set points is studied and in Rao and Rawlings (1999), the target calculation problem. Disturbance models are widely analyzed in Pannocchia and Rawlings (2003), Pannocchia and Bemporad (2007), Maeder et al. (2009), and Maeder and Morari (2010). Another offset-free MPC based on the internal model principle can be found in Magni and Scattolini (2007). Further results on MPC for tracking are addressed in Ferramosca et al. (2009). A survey on the MPC for tracking can be found in Limon et al. (2012).

Bibliography

Angeli D, Amrit R, Rawlings JB (2012) On average performance and stability of economic model predictive control. IEEE Trans Autom Control 57: 1615–1626

Bemporad A, Casavola A, Mosca E (1997) Nonlinear control of constrained linear systems via predictive reference management. IEEE Trans Autom Control 42:340–349

Camacho EF, Bordons C (2004) Model predictive control, 2nd edn. Springer-Verlag, London

Chisci L, Zappa G (2003) Dual mode predictive tracking of piecewise constant references for constrained linear systems. Int J Control 76:61–72

Ferramosca A, Limon D, Alvarado I, Alamo T, Camacho EF (2009) MPC for tracking with optimal closed-loop performance. Automatica 45:1975–1978

Findeisen R, Chen H, Allgöwer F (2000) Nonlinear predictive control for setpoint families. In: Proceedings of the American control conference, Chicago, USA, pp 260–264

Limon D, Alvarado I, Alamo T, Camacho EF (2008) MPC for tracking of piece-wise constant references for constrained linear systems. Automatica 44:2382–2387

Limon D, Ferramosca A, Alamo T, Gonzalez AH (2012) Model predictive control for changing economic targets. Paper presented at the IFAC conference on nonlinear model predictive control 2012 (NMPC'12), Noordwijkerhout, 23–27 Aug 2012

Maeder U, Morari M (2010) Offset-free reference tracking with model predictive control. Automatica 46(9):1469–1476

Maeder U, Borrelli F, Morari M (2009) Linear offset-free model predictive control. Automatica 45:2214–2222

Magni L, Scattolini R (2005) On the solution of the tracking problem for non-linear systems with MPC. Int J Syst Sci 36(8):477–484

Magni L, Scattolini R (2007) Tracking on non-square nonlinear continuous time systems with piecewise constant model predictive control. J Process Control 17:631–640

Magni L, De Nicolao G, Scattolini R (2001) Output feedback and tracking of nonlinear systems with model predictive control. Automatica 37:1601–1607

Muske K (1997) Steady-state target optimization in linear model predictive control. Paper presented at the 16th American control conference, Albuquerque, 4–6 June 1997

Olaru S, Dumur D (2005) Compact explicit MPC with guarantee of feasibility for tracking. In: Conference decision and control and European control conference 2005, Seville, Spain, pp 969–974

Pannocchia G (2004) Robust model predictive control with guaranteed setpoint tracking. J Process Control 14:927–937

Pannocchia G, Bemporad A (2007) Combined design of disturbance model and observer for offset-free model predictive control. IEEE Trans Autom Control 52:1048–1053

Pannocchia G, Kerrigan E (2005) Offset-free receding horizon control of constrained linear systems. AIChE J 51:3134–3146

Pannocchia G, Rawlings JB (2003) Disturbance models for offset-free model-predictive control. AIChE J 49:426–437

Rao CV, Rawlings JB (1999) Steady states and constraints in model predictive control. AIChE J 45:1266–1278

Rawlings JB, Mayne DQ (2009) Model predictive control: theory and design, 1st edn. Nob-Hill Publishing, Madison

Rossiter JA, Kouvaritakis B, Gossner JR (1996) Guaranteeing feasibility in constrained stable generalized predictive control. IEEE Proc Control Theory Appl 143:463–469

Wan Z, Kothare MV (2003) An efficient off-line formulation of robust model predictive control using linear matrix inequalities. Automatica 39:837–846

Transmission

Luigi Iannelli
Università degli Studi del Sannio, Benevento, Italy

Abstract

Automotive transmissions are fundamental components in modern vehicles. They are required to make the engine operating at the most efficient operating point for providing the necessary torque at the wheels and minimizing the fuel consumptions. Moreover, transmissions should be able to smooth or to filter out power source torque oscillations that can appear in the driveline. For achieving such objectives, the automotive industry has looked at different technological solutions. The introduction of electronically controlled transmissions contributed to augment the possibilities of new solutions that would not have been implementable without the flexibility and the performance of electronic control. Thus, recent technological developments of automotive transmissions gave the opportunity to engineers and control scientists for investigating challenging control problems with daily life practical applications.

Keywords

Actuators; Clutch engagement; Driveline; Dry clutch; Electrohydraulic; Electronic control; Gear shifting; Multivariable control; Optimal control; Powertrain; Sensors; Smart materials; Torque converter

Introduction

In motor vehicles, the transmission is an important system that transfers the power generated by the *internal combustion engine* to the wheels, according to the driver's requests. The transmission, together with the engine, the driveshaft, differential, and driven wheels, constitutes the *powertrain* (sometimes *driveline* or *drive train* is used to denote the powertrain excluding the engine and the transmission). The first fundamental objective of a transmission is to adjust the ratio between the wheel speed and the engine speed in order to achieve the optimal operating point of the engine, independently of the vehicle velocity. Indeed, typical internal combustion engines provide low torques at low engine speeds, and, thus, it is necessary to amplify the torque making the engine work at higher speeds when the vehicle is at low speeds, e.g., during a launch from standstill. From an equivalent point of view, the transmission allows to amplify the engine

torque transferred to the wheels when higher accelerations are needed. For such reasons, every type of transmission has some devices that allow selection of select the ratio between its input shaft angular speed (engine side) and its output shaft angular speed (the side toward the wheels). The transmission's input shaft is connected to the *flywheel* of the engine, while the output shaft of the transmission is connected to the *final drive* (containing the *differentials*) through the *drive shaft*. (In British English, the term propeller shaft is also used when dealing with a rear-wheel-driven vehicle.) Even though such subsystems are differently located depending on the vehicle layout (if front-wheel or rear-wheel driven or even all-wheel driven), they all are parts of the powertrain and determine its behavior.

Transmissions can be viewed also as systems that allow the transfer of power from the engine to the vehicle in a smooth and efficient way. In order to achieve such basic and fundamental objectives as well as to improve fuel economy, performance, and drivability, many technologies have been introduced into the market of automotive transmissions.

Types of Transmissions

In **manual transmissions** (MT), a set of gears provides each the different speed conversion ratios, and any gear can be selected by the driver by acting on the shift lever. For interrupting the power flow during the gear selection, a *clutch* is requested that disconnects the transmission from the flywheel of the engine and reconnects it just after the selection of the new gear. All such operations are performed manually by the driver.

Automatic transmissions (AT) are the other well-known type of automotive transmissions. Hydraulic ATs do not have the clutch for connecting the transmission to the flywheel; instead they have a *torque converter* that provides the fluid coupling between the transmission and the engine realizing both a damping of the powertrain vibrations and a torque multiplication. Moreover in ATs, a set of *planetary gear* allows the selection

of different gear ratios. The driver selects only the operation mode, and the selection of the gear is implemented through the electronic control. The main limits of these transmissions are the low efficiency (particularly due to the slip of the torque converter), a larger space requirement, and a higher weight. The new generations of automatic transmissions have reduced such disadvantages, thanks to the use of lightweight materials and more shifting steps, and, above all, the replacement of conventional hydraulic components by electronic and electrohydraulic counterparts.

Indeed the electronic transmission control not only improves fuel economy, performances, and drivability, but it also gives flexibility and new possibilities (e.g., diagnostics, fault detection, integration with other subsystems) that overcome the intrinsic disadvantages like complexity and development cost (Deur et al. 2006). Electronic transmission control played a fundamental role also in the introduction of new technologies that have exploited automatic control techniques. Thus, in recent years, continuously variable transmissions, automated manual transmissions, dual clutch transmissions, and electrically variable transmissions have appeared in the market, which was traditionally dominated by ATs and MTs (Sun and Hebbale 2005).

An **automated manual transmission** (AMT) system can be viewed as an MT with some controlled actuators as add-ons: it still has a (dry or wet) clutch assembly and a multispeed gearbox, both of which are equipped with electromechanical or electrohydraulic actuators which are commanded by an electronic control unit. In AMTs the gearshift can be decided automatically by the transmission control unit (TCU) or even manually by the driver. In both cases after the gearshift command, the TCU manages all the shifting steps, through suitable signals sent to the engine, the clutch assembly, and the gearbox. This technology has the advantages of lower weight, lower costs, and higher efficiency with respect to ATs.

It is worth highlighting one limitation of AMTs, the reduction in driving comfort caused by lack of traction during gear shift actuation.

T

Indeed the torque interruption leads to perceived jerks due to vehicle acceleration discontinuity and is very different compared to the smoother conventional automatic transmissions with torque converters (Lucente et al. 2007).

Automated manual transmissions have become popular in Europe for their higher performances with respect to MTs and for their lower cost compared to ATs. In North America, instead, their use is limited because of the torque interruption during shifts that causes some discomfort.

An offshoot of the AMT is the **dual clutch transmission** (DCT), in which the gearbox assembly has two separate and independent clutches, one for odd gears and one for even gears. In a DCT, shifts can be achieved without noticeable torque gap, by applying the engine torque to one clutch while the engine torque is being disconnected from the other clutch. The result is gentle, jerk-free gear shifts with the same comfortable driving of an automatic transmission combined with the efficiency and the performance of an economic manual transmission. In both DCT and AMT, electronic control (in particular aimed to solve the clutch engagement control problem) is the key to ensuring a smooth torque transfer.

As a further transmission technology, the **continuously variable transmission** (CVT) enables the engine to operate in a wide range of speed and load conditions independently from the speed and the torque requests of the vehicle. A modern CVT system consists of a steel belt that runs between two variable-width pulleys. The distance between pulley cones can be varied to change the gear ratio between shafts, thus generating an infinite number of "gears." A CVT is less efficient than a standard discrete AT due to the losses in the belt-pulley system, but it can improve fuel economy by making the engine work in better operating conditions. Related to CVT is the **electrically variable transmission** (EVT) that appeared in the market with hybrid electric vehicles recently and use electric machines, namely motors/generators with planetary gear sets, so as to enable the function of CVTs with flexibility, controllability, and better performance. This type of transmission is usually found in hybrid electric vehicles.

By looking at the different types of automotive transmissions, it is possible to classify electronically controlled transmissions into two groups: discrete ratio and continuously variable transmissions. The first group deals with the problem of automating the shift scheduling ("when-to") or also controlling the shift execution ("how-to") (Hrovat and Powers 1988). The latter group deals with control problems that live in a continuous domain (like classical "process control"), and that allows the design of simpler control software. Indeed the discrete ratio transmissions are more complex to control since they determine many large transients of short duration due to gear shifts, and moreover their intrinsic mixed discrete-continuous nature gives rise to dynamic systems in which continuous time dynamics interact with discrete event dynamics. In other words such class of controlled transmissions can be considered a significant application of what control theory calls **hybrid systems**. This makes transmission control a very interesting and challenging control engineering problem.

Control Problems for Automotive Transmissions

Electronic control applied to automotive transmissions enables improved efficiency and fuel economy, better shift quality and comfort, and flexible driving. In order to achieve such objectives, different approaches can be used for designing suitable control laws, and a hierarchical approach is often required for dealing with the complexity of the several problems. Thus, recently produced cars together with the engine control unit also have a TCU that manages all transmission operations and sends command signals to actuators in order to perform the desired behavior. Some dedicated devices are then available for tackling challenging control problems and for trying to solve them exploiting classical feedback and/or modern model-based control.

Low Level Control

In electronically controlled transmissions, many hydraulic functions of conventional transmissions have to be replaced by electrohydraulic systems. Thus it is a fundamental requirement to be able to control actuators in a suitable way. Usually classical PID regulators are employed for this type of low level controls whose aim consists of regulating some variables to reference values computed by a higher level controller. For instance, in AMTs the concentric slave cylinder is controlled to make the clutch disk follow a position reference signal computed by the TCU. The clutch position reference can be obtained by taking into account some models of the clutch transmission characteristic (Vasca et al. 2011), thus realizing a feedforward/feedback architecture.

Other examples of low level controls in automotive transmissions are related to the clutch fill process (Song et al. 2010) in ATs, or the line pressure control, or also the CVT belt load control.

Calibration Process

Most of the industrial control strategies applied to automotive transmissions are based on feedforward/feedback architectures. Feedforward control typically relies on detailed models of the transmission that quite often consist of some lookup tables rather than specific physical models. Lookup tables are used also for implementing adaptive feedback controllers, and thus, the use of tables with calibrated variables is widespread in automotive transmission control. With the increasing number of required functionalities, the calibration process for transmission control subsystems becomes more and more complex. Then it becomes important to investigate automated and systematic approaches for the calibration process in order to improve the reliability and performances and, above all, for diminishing the development time.

Of course, a different approach that looks at developing model-based control strategies could be the way to reduce the number of calibration variables and, thus, the calibration effort and time. The main obstacle to that is the uncertain environment that makes it very challenging to find robust control solutions.

Gear Shifting

The gear shift execution is a common problem in all discrete ratio transmission. In Figs. 1 and 2, the schemes of two transmission architectures are reported. Although we are looking at completely different typologies like ATs and AMTs, the gear shift problem is almost the same from an abstract point of view: commanding the actuator for getting a desired torque at the primary shaft of the transmission in order to have the possibility of disengaging the old gear, engaging the neutral gear, and then engaging the new gear, without shuffles, and limiting the jerk experienced by the driver. In ATs the basic idea is to control the hydraulic pressures of the torque converter to transfer smoothly the power from the engine to the driveline while minimizing the torque disturbance at the output shaft. Analogously in AMTs with dry clutch, the actuator is commanded for positioning the clutch disk toward the flywheel, exerting a pressure that is transformed into the transmitted torque.

For instance, a wet clutch of an automatic transmission gives the following transmitted torque (Deur et al. 2006):

$$T = n \, A_p \, p_{\text{app}} \, \mu(\omega, p_{\text{app}}, \theta) \, r_e \, \text{sgn}(\omega)$$

where n is the number of friction surfaces, A_p is the piston area, p_{app} is the hydraulic pressure, μ is the friction coefficient (depending also on the clutch fluid temperature θ), r_e is the equivalent radius of the clutch, and ω is the clutch slip speed, i.e., the difference between the engine speed and the speed of the input shaft of the transmission.

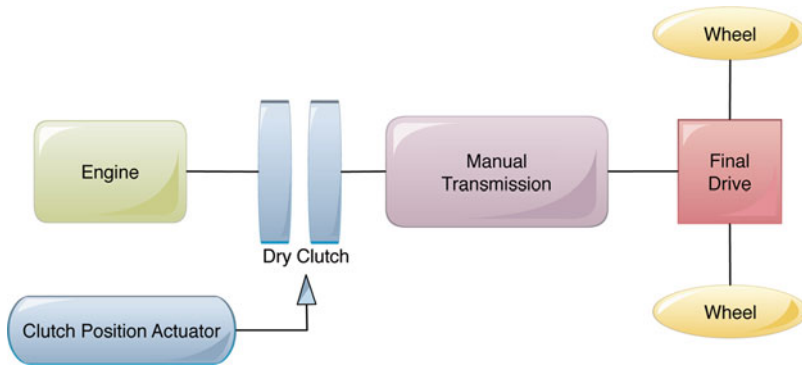For a dry clutch of an automated manual transmission (Vasca et al. 2011),

$$T = nF_{pp}(x_{to})\mu(\omega, \theta)r_e \, \text{sgn}(\omega)$$

where F_{pp} is the force exerted by the cushion spring depending on the clutch actuator position x_{to} and θ is the clutch disk temperature.

In both cases, the actuator allows regulation of the torque transmitted, respectively, through the
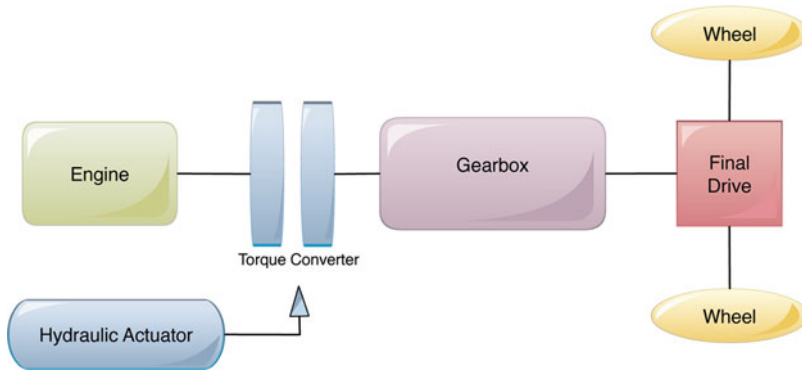
Transmission, Fig. 1 Architecture of an automatic transmission

Transmission, Fig. 2 Architecture of an automated manual transmission

torque converter or the dry clutch under a slipping condition. The main problem is that in modern transmissions, there are no low-cost torque sensors, and thus, due also to model uncertainties and highly variable operating conditions, it is not possible to regulate the transmitted torque through a closed-loop scheme. What is usually done is to control the engine speed and/or the speed of the input shaft of the transmission. Quite often, their difference (the slip speed) is the variable to be controlled.

Many different approaches have been proposed in the literature for solving such control problems that can be formulated as simply as a regulation problem of the slip speed or as a more complex multivariable control problem that considers the engine and clutch torques as control variables and the slip speed and vehicle speed as controlled variables, possibly solving the problem through robust control tools. The

problem can be formulated quite naturally also as an optimal control problem that aims to minimize the engagement time, the driveline oscillations, or the dissipated energy. For example, by defining the time derivative of the clutch torque as one control variable, the transmitted torque becomes a state variable, and the energy dissipated during the engagement phase can be expressed as the cross product of two state variables (Garofalo et al. 2002)

$$E_d = \int_0^{\bar{t}} \omega(s) T(s) \mathrm{d}s,$$

and the clutch engagement can be expressed as an optimal control problem with free final time (the engagement time, $\bar{t}$) and a final state constraint (i.e., $\omega(\bar{t}) = 0$).

Some authors have also proposed a different solution for the gear shifting problem by

acting through the engine control (Pettersson and Nielsen 2000). The idea is to control the gear shifting using directly the engine as the actuator that allows to modulate the transferred torque to the transmission (see Figs. 1 and 2). In particular the engine is controlled so as to get a zero transferred torque in the transmission and then the neutral gear is engaged. In this way, a virtual clutch is realized.

Driveline Modeling

When model-based control is used for automotive transmissions, it is important to have a good model of the driveline that is detailed enough for capturing the main dynamics and, at the same time, sufficiently simple to deal with for designing not so complex controllers. Vehicular drivelines have many elastic parts making mechanical resonances occurring. Handling such resonances is important for driveability but also for reducing mechanical stresses. Thus driveline control is crucial not only during gear shifting but also for a more general powertrain control that could manage wheel-speed oscillations induced by sudden accelerations or following from the road roughness.

Integrated Powertrain Management

Shift scheduling is an additional interesting problem of electronically controlled transmissions. The shift point is usually based on some measurements like the vehicle speed, the maximum acceleration or throttle angle. In this case the control strategy is open-loop and implemented through lookup tables.

As the number of gears increases, the shift schedule gets more complicated. Thus it becomes important to take into account also the actual driving scenario. For example, entering a curve during uphill driving is quite different relative to a downhill driving situation, so it can be very useful getting information on the steering angle, the road grade, vehicle acceleration, etc. More information, together with new degrees of freedom that are available to modern vehicles (e.g., vehicles with electronic throttle control), give the opportunity to realize an integrated powertrain control which coordinates the engine control and the transmission control allowing to manage the gear scheduling and the gear shifting execution in a more flexible way, trying to optimize the fuel consumption and to improve the driveability (Kim et al. 2007).

Diagnostics

In all automotive applications, safety is a fundamental issue that becomes more and more critical when the number of subsystems and their interaction increases, as it happens when introducing electronic control. Thus, diagnosing faults of control systems is a challenging problem, in particular when there is limited information. To this aim, systems and control theory can be very useful for designing observer or fault detection algorithms that could deal with these types of problems.

Summary and Future Directions

In summary transmission control is a fertile application for looking at challenging problems of much interest for both control scientists and engineers, giving the opportunity for investigating topics like optimal control (e.g., for gear shifting and integrated powertrain control), robust control (for driveline modeling and control), estimation (diagnostics), and adaptive and predictive control.

Technological developments could affect the possibilities and the effectiveness of transmission control. Of course new sensing devices can improve the reliability and the precision of the feedback, but they can also open the door to new control architectures. For instance, the phenomenon of inverse magnetostriction that converts material strain into magnetic property changes can be exploited to measure transmitted torque, and some magnetoelastic torque sensors have been investigated by a number of researchers (Klimartin 2003; Pietron et al. 2013). In this way the gear shifting control problem can be attacked by closing the loop on the transmitted torque measurement, avoiding more or less complex torque observers or, at least, improving the control performances.

Analogous considerations can be carried out at the actuation level. For instance, Kim and Choi (2011) have proposed a new clutch actuator with a self-energizing mechanism so as to amplify the normal force applied on the contact surfaces for the engagement. That idea allows the clutch module to consume less energy for actuating the overall system. Smart-material-based actuation devices were also developed by a number of researchers in recent years (Chaudhuri and Wereley 2012), and some specific applications to automotive transmissions are currently under investigation, like magnetorheological fluid dual clutch transmissions that are discussed in Chen et al. (2012).

At the system level, one of the most interesting research directions deals with the communication and coordination among different control subsystems like the engine control, transmission control, and electronic stability control, with the final aim of integrating all such subsystems for integrated powertrain management.

Cross-References

▶ Engine Control
▶ Powertrain Control for Hybrid-Electric and Electric Vehicles

Recommended Reading

Hrovat et al. (2010) give an overview of automotive transmissions in their chapter on powertrain control of the CRC Control Handbook. Of course transmission control is discussed also in classical automotive control books: one of the first well-known books dedicated to automotive control was Kiencke and Nielsen (2005). There a whole chapter on driveline control deals with driveline modeling and gear shifting for clutch-based transmissions. A more recent book on automotive control is Ulsoy et al. (2012) where transmission control for all-wheel drive vehicles is also presented.

In the scientific literature, many papers deal with transmission control: here, in particular, we would like to cite the optimal control approach for ATs by Haj-Fraj and Pfeiffer (2001), a deep discussion on AMT control in Glielmo et al. (2006), and, more recently, papers on DCTs like Kulkarni et al. (2007) and Senatore (2009); in the latter, the author illustrates the wide selection of patents on dual clutch.

Regarding automotive technologies, an overview of automotive sensors can be found in Fleming (2008), while a discussion on smart materials and the integration of mechanics, materials, and electronics (the so-called mechamatronics discipline) are presented by Munhoz et al. (2007).

Bibliography

Chaudhuri A, Wereley N (2012) Compact hybrid electrohydraulic actuators using smart materials: a review. J Intell Mater Syst Struct 23(6):597–634

Chen D, Xu J, Pan J, Guo X, Sun W (2012) Research and prospect of automobile magneto-rheological fluid dual clutch transmission. In: Proceedings of the 2nd international conference on electronic & mechanical engineering and information technology, Shenyang, China, pp 98–102

Deur J, Petric J, Asgari J, Hrovat D (2006) Recent advances in control-oriented modeling of automotive power train dynamics. IEEE/ASME Trans Mechatron 11(5):513–523

Fleming WJ (2008) New automotive sensors—a review. IEEE Sens J 8(11):1900–1921

Garofalo F, Glielmo L, Iannelli L, Vasca F (2002) Optimal tracking for automotive dry clutch engagement. In: 15th IFAC world congress, Barcelona, pp 367–372

Glielmo L, Iannelli L, Vacca V, Vasca F (2006) Gearshift control for automated manual transmissions. IEEE/ASME Trans Mechatron 11(1):17–26

Haj-Fraj A, Pfeiffer F (2001) Optimal control of gear shift operations in automatic transmissions. J Frankl Inst 338:371–390

Hrovat D, Powers W (1988) Computer control systems for automotive power trains. IEEE Control Syst Mag 8(4):3–10

Hrovat D, Jankovic M, Kolmanovsky I, Magner S, Yanakiev D (2010) Powertrain control. In: The control handbook. Control applications, 2nd edn. CRC, Boca Raton

Kiencke U, Nielsen L (2005) Automotive control systems: for engine, driveline, and vehicle. Springer, Berlin/Heidelberg

Kim J, Choi SB (2011) Design and modeling of a clutch actuator system with self-energizing mechanism. IEEE/ASME Trans Mechatron 16(5):953–966

Kim D, Peng H, Bai S, Maguire JM (2007) Control of integrated powertrain with electronic throttle and automatic transmission. IEEE Trans Control Syst Technol 15(3):474–482

Klimartin B (2003) Magnetoelastic torque sensor utilizing a thermal sprayed sense-element for automotive transmission applications. SAE technical paper 2003-01-0711

Kulkarni M, Shim T, Zhang Y (2007) Shift dynamics and control of dual-clutch transmissions. Mech Mach Theory 42(2):168–182

Lucente G, Montanari M, Rossi C (2007) Modelling of an automated manual transmission system. Mechatronics 17(2–3):73–91

Munhoz D, Gregolin J, de Faria L, de Andrade T (2007) Automotive materials: current status, technology trends and challenges. SAE technical paper 2007-01-2671

Pettersson M, Nielsen L (2000) Gear shifting by engine control. IEEE Trans Control Syst Technol 8(3):495–507

Pietron G, Fujii Y, Kucharski J, Yanakiev D, Kapas N, Hermann S, Hogirala R, Green T (2013) Development of magneto-elastic torque sensor for automatic transmission applications. SAE Int J Passeng Cars Mech Syst 6(2):529–534

Senatore A (2009) Advances in the automotive systems: an overview of dual-clutch transmissions. Recent Pat Mech Eng 2(2):93–101

Song X, Zulkefli MAM, Sun Z (2010) Automotive transmission clutch fill optimal control: an experimental investigation. In: Proceedings of the American control conference, Baltimore. IEEE, pp 2748–2753

Sun Z, Hebbale K (2005) Challenges and opportunities in automotive transmission control. In: Proceedings of the American control conference, Portland, pp 3284–3289

Ulsoy AG, Peng H, Çakmakci M (2012) Automotive control systems. Cambridge University Press, Cambridge

Vasca F, Iannelli L, Senatore A, Reale G (2011) Torque transmissibility assessment for automotive dry-clutch engagement. IEEE/ASME Trans Mechatron 16(3):564–573

T

Uncertainty and Robustness in Dynamic Vision

Mario Sznaier and Octavia Camps
Electrical and Computer Engineering
Department, Northeastern University,
Boston, MA, USA

Abstract

Dynamic vision is a subfield of computer vision dealing explicitly with problems characterized by image features that evolve in time according to some underlying dynamics. Examples include sustained target tracking, activity classification from video sequences, and recovering 3D geometry from 2D video data. This article discusses the central role that systems theory can play in developing a robust dynamic vision framework, ultimately leading to vision-based systems with enhanced autonomy, capable of operating in stochastic, cluttered environments.

Keywords

Event detection; Multiframe tracking; Structure from motion

Background

In this article, we represent linear time invariant (LTI) systems by their associated transfer matrix $G(z)$. The "size" of $G(z)$, which plays a key role in assessing the effects of uncertainty, will be measured using the $\mathcal{H}_\infty$ norm, defined as $\|G\|_\infty \doteq \sup_\omega \overline{\sigma}\left(G(e^{j\omega})\right)$, where $\overline{\sigma}(.)$ denotes maximum singular value. For scalar systems, this reduces to the peak value of the frequency response (i.e., the maximum gain of the system). In the matrix case, this definition takes into account both the worst-case frequency and spatial direction. Background material on the $\mathcal{H}_\infty$ norm, its computation and its significance in the context of robust control theory, is given in Sánchez–Peña and Sznaier (1998). A general coverage of linear systems theory, including alternative representations of linear systems and their associated properties, can be found, for instance, in Rugh (1996).

Multiframe Tracking

A requirement common to most dynamic vision applications is the *ability to track* objects across frames, in order to collect the data required by a subsequent activity analysis step. Current approaches integrate correspondences between individual frames over time, using a combination of some assumed simple target dynamics (e.g., constant velocity) and empirically learned noise distributions (Isard and Blake 1998; North et al. 2000). However, while successful in many scenarios, these approaches are vulnerable to model uncertainty, occlusion, and appearance changes, as illustrated in Fig. 1.
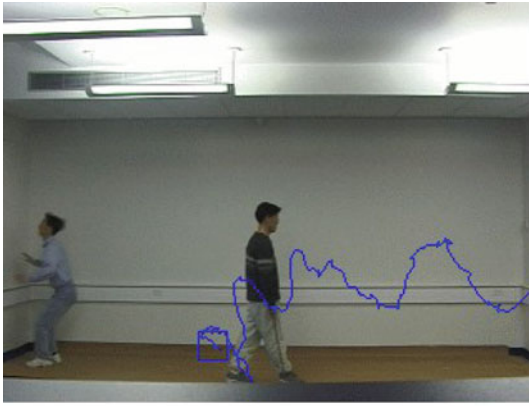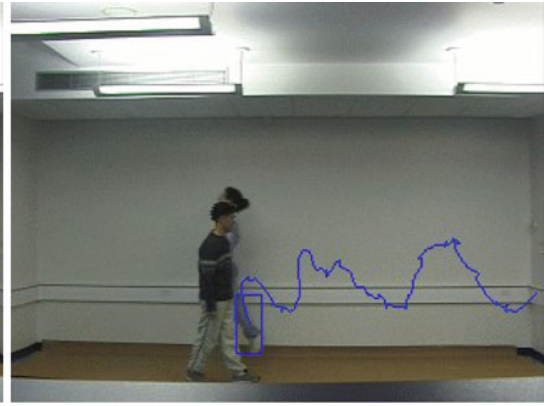
Frame 150   Frame 95

Uncertainty and Robustness in Dynamic Vision, Fig. 1 Unscented particle filter-based tracking in the presence of occlusion

As shown next, the fragility noted above can be avoided by modeling the motion of the target as the output of a dynamical system, to be identified directly from the available data, along with bounds on the identification error. In the sequel, we consider two different cases: (i) the motion of the target is known to belong to a relatively small set of a priori known motion modalities; and (ii) no prior knowledge is available.

The case of known motion models: Consider first the case where a set of models known to span all possible motions of the target is known a priori, as it is often the case with human motion. In this case, the position y_k of a given target can be modeled as $y(z) = \mathcal{F}(z)e(z) + \eta(z)$ where $\mathbf{e}$ and η_k denote a suitable input and measurement noise, respectively, and where $\mathcal{F}$ admits an expansion of the form $\mathcal{F} = \overbrace{\sum_{j=1}^{N_p} p_j \mathcal{F}^j}^{\mathcal{F}_p} + \mathcal{F}_{np}$. Here $\mathcal{F}^j$ represent the (known) motion modalities of the target and $\|\mathcal{F}_{np}\|_\infty \leq K$, e.g., a bound on the maximum admissible approximation error of the expansion $\mathcal{F}_p$ to $\mathcal{F}$ is available. In the reminder of this article, we will further assume that a set membership descriptions $\eta_k \in \mathcal{N}$ is available and, without loss of generality, that $e(z) = 1$ (i.e., motion of the target is modeled as the impulse response of the unknown operator $\mathcal{F}$).

In this context, the next location of the target feature y_k can be predicted by first identifying the relevant dynamics $\mathcal{F}$ and then using it to propagate its past values. In turn, identifying the dynamics entails finding an operator $\mathcal{F}(z) \in \mathcal{S} \doteq \{\mathcal{F}(z): \mathcal{F} = \mathcal{F}_p + \mathcal{F}_{np}\}$ such that $y - \eta = \mathcal{F}$, precisely the class of interpolation problem addressed in Parrilo et al. (1999). As shown there, finding such an operator reduces to solving a linear matrix inequality (LMI) feasibility problem. Once this operator is found, it can be used in conjunction with a particle (or a Kalman) filter to predict the future location of the target. Figure 2 shows the tracking results obtained using this approach. Here, we used a combination of a priori information: (i) 5 % noise level and (ii) $\mathcal{F}_p \in \text{span}[\frac{1}{z-1}, \frac{z}{z-a}, \frac{z}{(z-1)^2}, \frac{z^2}{(z-1)^2}, \frac{z^2-\cos\omega z}{z^2-2\cos\omega z+1}, \frac{\sin\omega z^2}{z^2-2\cos\omega z+1}]$ where $a \in \{0.9, 1, 1.2, 1.3, 2\}$ and $\omega \in \{0.2, 0.45\}$. The experimental information consisted of the position of the target in $N = 20$ frames, where it was not occluded. Note that, by exploiting predictive power of the identified model, the Kalman filter is now able to track the target past the occlusion, eliminating the need for using a (more computationally expensive) particle filter.

Unknown motion models: This case could be addressed in principle by performing a purely nonparametric worst-case identification (Parrilo et al. 1999) and then proceeding as above.
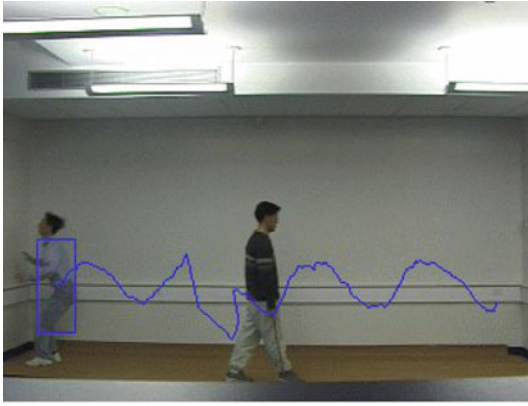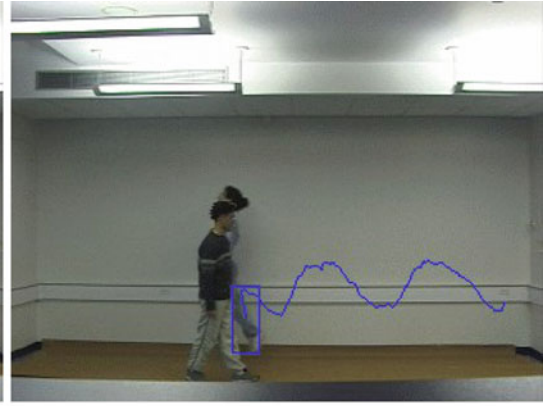
Uncertainty and Robustness in Dynamic Vision, Fig. 2 Using the identified model in combination with Kalman filter allows for robust tracking in the presence of occlusion

However, a potential difficulty here stems from the high order of the resulting model (recall that the order of the central interpolant is the number of experimental data points). If a bound n on the order of the underlying models is available, this difficulty can be avoided by recasting the prediction problem into a rank minimization form, which in turn can be relaxed to a semi-definite optimization. To this effect, recall that (Ding et al. 2008), in the absence of noise, given $2n$ values of $\{\mathbf{y}_k\}_{k=t-2n+1}^{t}$, its next value $\mathbf{y}_{t+1}$ is the unique solution to the following rank minimization problem:

$$\mathbf{y}_{t+1} = \underset{\mathbf{y}}{\mathrm{argmin}}\{\mathrm{rank}\,[\mathbf{H}_{n+1}(\mathbf{y})]\} \text{ where } \mathbf{H}_{n+1}(\mathbf{y})$$

$$\doteq \begin{bmatrix} \mathbf{y}_{t-2n+1} & \mathbf{y}_{t-2n+2} & \cdots & \mathbf{y}_{t-n} \\ \mathbf{y}_{t-2n+2} & \mathbf{y}_{t-2n+3} & \cdots & \mathbf{y}_{t-n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{t-n+1} & \mathbf{y}_{t-n+2} & \cdots & \mathbf{y} \end{bmatrix} \quad (1)$$

Clearly, the same result holds if multiple elements of the sequence $\mathbf{y}$ are missing, at the price of considering longer sequences (the total number of data points should exceed $2n$). This result allows for handling both noisy and missing data (due, for instance, to occlusion), by simply solving

$$\min_{\boldsymbol{\zeta}} \{\mathrm{rank}\,[\mathbf{H}(\boldsymbol{\zeta})]\} \text{ subject to } \mathbf{v} \in \mathcal{N}_v$$
$$\text{where } \zeta_i = \begin{cases} \mathbf{y}_i - \mathbf{v}_i & \text{if } i \in \mathcal{I}_a \\ \mathbf{x}_i & \text{if } i \in \mathcal{I}_m \end{cases}$$
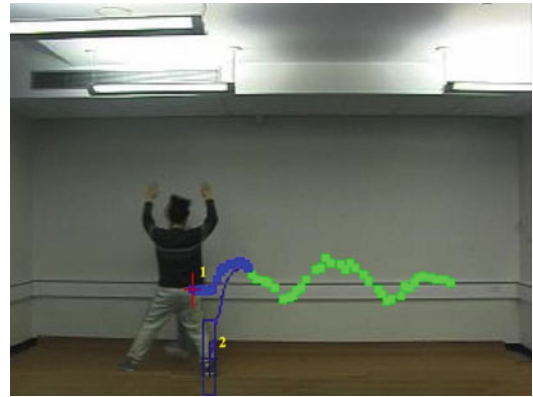
Uncertainty and Robustness in Dynamic Vision, Fig. 3 Trajectory prediction. Rank minimization (*1*) versus Kalman filtering (*2*)

$\mathcal{I}_a$ and $\mathcal{I}_m$ denote the set of available (but noisy) and missing measurements, respectively, and where $\mathcal{N}_v$ is a set membership description of the noise $\mathbf{v}$. In the case where $\mathcal{N}_v$ admits a convex description, using the nuclear norm as a surrogate for rank (Fazel et al. 2003) allows for reducing this problem to a convex semi-definite program. Examples of these descriptions are balls in ℓ_∞, e.g., $\mathcal{N} \doteq \{v : |v_k| \leq \epsilon\}$ or constraints on the norm of $\mathbf{H}_v$, the Hankel matrix of the noise sequence, which under mild ergodicity assumptions are equivalent to constraints on the magnitude of the noise covariance. Figure 3 illustrates the effectiveness of this approach.

U

As shown there, the rank minimization-based filter successfully predicts the location of the target, while a Kalman filter-based tracker fails due to the substantial occlusion.

Event Detection and Activity Classification

Using the trajectories generated by the tracking step for activity recognition entails (i) segmenting the data into homogeneous segments each corresponding to a single activity and (ii) classifying these activities, typically based on exemplars from a database of known activities. As shown in the sequel, both steps can be efficiently accomplished by exploiting the properties of the underlying system. The starting point is to model these activities as the output of a switched piecewise linear system. In this context, under suitable dwell time constraints, each switch (indicating a change in the underlying activity) can be identified by simply searching for points associated with discontinuities in the rank of the associated Hankel matrix, as illustrated in Fig. 4. Further, in this framework, the problem of classifying each subactivity can be recast into the behavioral model (in)validation setup shown in Fig. 5. Here $y_i(.)$ represents the impulse response of the (unknown) LTI system G, affected by measurement noise $\eta_i \in \mathcal{N}$ and uncertainty

$\Delta_i \in \mathcal{D}$ that accounts for the variability intrinsic to two different realizations of the same activity. Two different time series are considered to be realizations of the same activity if there exists at least one pair $(\eta_1, \eta_2) \in \mathcal{N}^2$, one pair $(\Delta_1, \Delta_2) \in \mathcal{D}^2$, a LTI system G with McMillan degree at most n_G, and suitable initial conditions $\mathbf{x}_1$, $\mathbf{x}_2$ resulting in the observed data. Remarkably, this model (in)validation problem can be reduced to a rank minimization form. In the simpler case where $\Delta_i = 0$, the problem can be solved using the following algorithm (Sznaier and Camps 2011):

Next, consider the more realistic case where the trajectories are also affected by bounded model uncertainty Δ, $\|\Delta\|_\infty \leq \gamma$, where γ is given as part of a priori information. In this scenario, the internal signal z is given by $z(t) = \zeta(t) - \eta(t)$, $\eta \in \mathcal{N}$, where the signal ζ satisfies

$$y = (1 + \Delta) * \zeta, \text{ for some } \Delta \in \mathcal{D} \qquad (2)$$

where $*$ denotes convolution. Exploiting Theorem 2.3.6 in Chen and Gu (2000) leads to an LMI condition in the variables z, η, for feasibility of (2). Thus, the only modification to Algorithm 1 required to handle model uncertainty is to incorporate this additional (convex) constraint to the rank minimization problems. Table 1 shows the results of applying this approach to
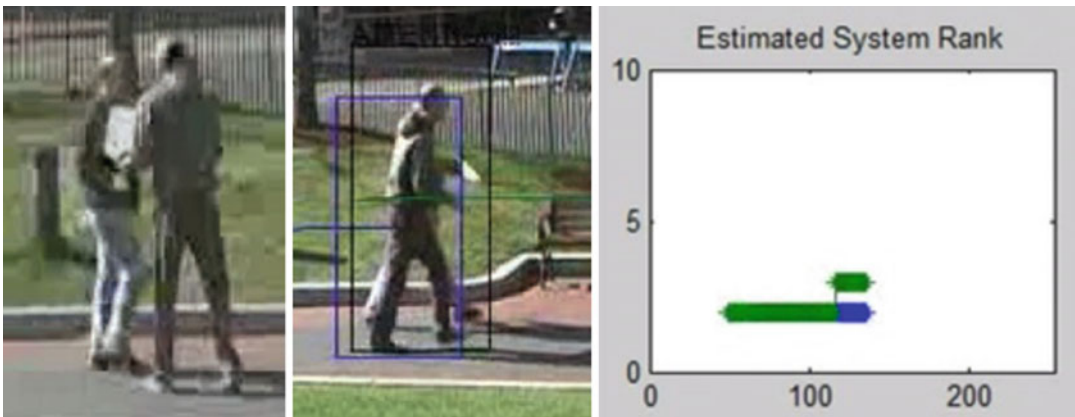
Uncertainty and Robustness in Dynamic Vision, Fig. 4 The jump in the rank of the Hankelmatrix corresponds to the time instant where the subjectsmeet and exchange a bag
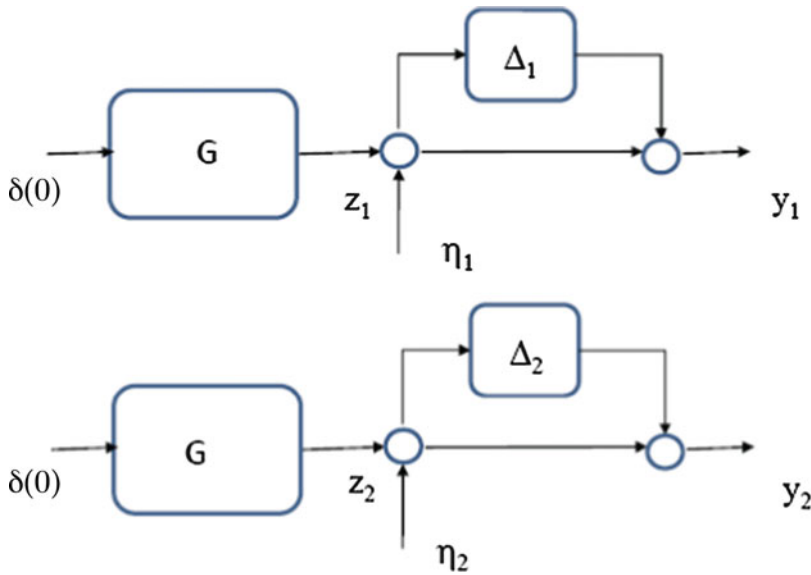
Uncertainty and Robustness in Dynamic Vision, Fig. 5 Model (in)validation setup

2 video sequences, walking and running, from the KTH database (Laptev et al. 2008). Sample frames from these sequences are shown in Fig. 6. In order to reduce the dimensionality of the data, the frames were first projected into a three-dimensional space using principal component analysis (PCA), and the resulting time series were used as the input to Algorithm 1, assuming 10 % noise and 10 % model uncertainty. As shown in Table 1, the algorithm correctly identifies the subsequences (a)–(c) as being generated by the same underlying activity (walking).

Uncertainty and Robustness in Dynamic Vision, Table 1 Activity classification results. Sequences (a)–(c) correspond to walking and (d) to running

Activity pair	Rank($\mathbf{H}_1$)	Rank($\mathbf{H}_2$)	Rank($[\mathbf{H}_1\ \mathbf{H}_2]$)
(a, b)	4	4	4
(a, c)	4	4	4
(a, d)	4	8	8

Algorithm 1 Behavioral model (in)validation

Data: Noisy measurements y_1, y_2.
A priori information: noise description $\eta_i \in \mathcal{N}$
1. Solve the following rank–minimization problems:
   $r_1^{min} = \min_{\eta_1} \text{rank}(\mathbf{H}_{y_1} - \mathbf{H}_{\eta_1})$
   subject to: $\eta_1 \in \mathcal{N}$.
   $r_2^{min} = \min_{\eta_2} \text{rank}(\mathbf{H}_{y_2} - \mathbf{H}_{\eta_2})$
   subject to: $\eta_2 \in \mathcal{N}$.
   $r_{12}^{min} = \min_{\eta_1} \text{rank}([\mathbf{H}_{y_{1n}}\quad \mathbf{H}_{y_{2n}}])$
   subject to: $\eta_1, \eta_2 \in \mathcal{N}$
   $\mathbf{H}_{y_{1n}} = \mathbf{H}_{y_1} - \mathbf{H}_{\eta_1}$
   $\mathbf{H}_{y_{2n}} = \mathbf{H}_{y_2} - \mathbf{H}_{\eta_2}$
2. The given trajectories were generated by the same LTI system with McMillan degree $\leq n_G$ iff:
   $r_1^{min} = r_2^{min} = r_{12}^{min} \leq n_G$

Summary and Future Directions

Vision-based systems are uniquely positioned to address the needs of a growing segment of the population. Aware sensors endowed with scene analysis capabilities can prevent crime, reduce time response to emergency scenes, and render viable the concept of ultra-sustainable buildings. Moreover, the investment required to accomplish these goals is relatively modest since a large number of cameras are already deployed and networked. Arguably, at this point, one of the critical factors limiting widespread use of these systems is their potential fragility when operating in unstructured scenarios. This article illustrates the key role that control theory can play in developing a comprehensive, provably robust

U

Uncertainty and Robustness in Dynamic Vision, Fig. 6 Sample frames from KTH activity video database. (**a**) Walking. (**b**) Running

dynamic vision framework. In turn, computer vision provides a rich environment both to draw inspiration from and to test new developments in systems theory.

details on the connection between identification and the problem of extracting actionable information from large data streams can be found, for instance, in Sznaier (2012).

Cross-References

▶ Particle Filters
▶ Estimation, Survey on

Recommended Reading

Details on how to select good features to track can be found in Richard Szeliski (2010). Using dynamics to recover 3D structure from 2D data is covered in Ayazoglu et al. (2010). Finally, further

Bibliography

Ayazoglu M, Sznaier M, Camps O (2010) Euclidean structure recovery from motion in perspective image sequences via Hankel rank minimization. LNCS, vol 6312. Springer, Berlin/New York, pp 71–84

Chen J, Gu G (2000) Control oriented system identification, An $\mathcal{H}_\infty$ approach. Wiley, New York

Ding T, Sznaier M, Camps O (2008) Receding horizon rank minimization based estimation with applications to visual tracking. In: Proceedings of the 47th IEEE conference on decision and control, Cancún, Dec 2008, pp 3446–3451

Fazel M, Hindi H, Boyd SP (2003) Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In: Proceedings of the 2003 ACC, Denver, pp 2156–2162

Isard M, Blake A (1998) CONDENSATION – conditional density propagation for visual tracking. Int J Comput Vis 29(1):5–28

Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: IEEE computer vision and pattern recognition, Anchorage, pp 1–8

North B, Blake A, Isard M, Rittscher J (2000) Learning and classification of complex dynamics. IEEE Trans Pattern Anal Mach Intell 22(9):1016–1034

Parrilo PA, Sánchez–Peña RS, Sznaier M (1999) A parametric extension of mixed time/frequency domain based robust identification. IEEE Trans Autom Control 44(2):364–369

Rugh WJ (1996) Linear systems theory, 2nd edn. Prentice Hall, Upper Saddle River

Sánchez–Peña RS, Sznaier M (1998) Robust systems theory and applications. Wiley, New York

Szeliski R (2010) Computer vision: algorithms and applications. Springer, New York

Sznaier M (2012) Compressive information extraction: a dynamical systems approach. In: Proceeding of the 2012 symposium on systems identification (SYSID 2012), July 2012, Brussels, pp 1559–1568

Sznaier M, Camps O (2011) A rank minimization approach to trajectory (in)validation. In: 2011 American control conference, pp 675–680

Underactuated Marine Control Systems

Kristin Y. Pettersen
Department of Engineering Cybernetics,
Norwegian University of Science and
Technology, Trondheim, Norway

Abstract

For underactuated marine vessels, the dimension of the configuration space exceeds that of the control input space. This article describes underactuated marine vessels and the control challenges they pose. In particular, there are two main approaches to design control systems for underactuated marine vessels. The first approach reduces the number of degrees of freedom (DOF) that it seeks to control such that the number of DOF equals the number of independent control inputs. The control problem is then a fully actuated control problem – something that simplifies the control design problem significantly – but special attention then has to be given to the inherent internal dynamics that has to be carefully analyzed. The other approach to design control systems for underactuated marine vessels seeks to control all DOF using only the limited number of control inputs available. The control problem is then an underactuated control problem and is quite challenging to solve. In this article, it is shown how line-of-sight methods can solve the underactuated control problems that arise from path following and maneuvering control of underactuated marine vessels.

Keywords

Marine vessels; Underactuated marine control problems; Underactuated marine vessels; Underactuation

Introduction

Marine systems are often equipped with fewer independent actuators than degrees of freedom. Examples include conventional ships/surface vessels that are typically equipped with a main thruster and a rudder or with two independent main thrusters, but without a side thruster. As a result, we have no control force in the sideways direction. This means that the forward motion (the surge motion) and the orientation (the yaw motion) can be controlled directly, while there is no direct way to influence the sideways motion of the surface vessel (the sway motion). The vessel is then said to be underactuated in sway. It is an underactuated system since it has only two independent control inputs, giving force and torque in surge and yaw, while the system has three degrees of freedom: surge, sway, and yaw. This underactuation leads to challenges when it comes to designing the control system.

Definition: Underactuated Marine Vessels

In order to properly define what we mean by underactuated marine vessels, we need the mathematical model (▶ Mathematical Models of Ships and Underwater Vehicles; Fossen 2011):

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \begin{bmatrix} \tau \\ 0 \end{bmatrix}$$

$$\dot{\eta} = J(\eta)v$$

where the configuration vector $\eta \in \mathbb{R}^n$, the velocity vector $v \in \mathbb{R}^n$, while the vector of independent control inputs $\tau \in \mathbb{R}^m$. The vessel is underactuated because $n > m$, i.e., the dimension of the configuration space exceeds that of the control input space (Oriolo and Nakamura 1991; Pettersen and Egeland 1996).

The underactuation leads to a second-order nonholonomic constraint

$$M_u\dot{v} + C_u(v)v + D_u(v)v + g_u(\eta) = 0$$

where M_u denotes the last $n - m$ rows of the matrix M and $C_u(v)$, $D_u(v)$, and $g_u(\eta)$ are defined similarly.

Definitions of nonholonomic and holonomic constraints can be found in Goldstein (1980). More facts about these kinds of constraints and conditions for when this second-order nonholonomic can be integrated to either a first-order nonholonomic or a holonomic constraint can be found in Tarn et al. (2003).

Control of Underactuated Marine Vessels

As we have seen above, the underactuation leads to a constraint, and this gives challenges when it comes to designing the control system. In particular, it can be shown that if $g_u(\eta)$ has a zero element, then there exists no continuous or discontinuous state feedback law that can asymptotically stabilize the equilibrium point $(\eta, v) = (0, 0)$ (Pettersen and Egeland 1996). This means

that in order to stabilize an equilibrium point, control methods from linear or classical nonlinear control theory cannot be applied.

There are two main classes of approaches to control underactuated marine vessels. The first class approaches the control problem by reducing the number of degrees of freedom that are to be controlled, while the other class seeks to control all degrees of freedom using the limited number of control inputs available.

If we reduce the number of degrees of freedom (DOF) that we seek to control, such that the number of DOF agrees with the number of independent control inputs, then we have a fully actuated control problem although the vessel is underactuated. This may at first sight look like a very simple way to design a control system for underactuated marine vessels. Note, however, that then, there will inherently be internal dynamics that needs to be examined carefully (Isidori 1995; Nijmeijer and van der Schaft 1990). Say, for instance, that we only care about controlling the position of the ship, and we choose not to care very much about the orientation of the ship. We do, for instance, want the ship to follow a straight line trajectory $(x_r(t), y_r(t))$, where x and y give the ship's position in an earth-fixed coordinate system, and the angle giving the ship orientation, ψ, is not so important to us. It is quite straightforward to use, for instance, output feedback linearization to this end (Isidori 1995; Nijmeijer and van der Schaft 1990). The resulting dynamics of the subsystem (x, y) is then called the external dynamics. We have full control over this using the two independent control inputs and can make it track any smooth trajectory $(x_r(t), y_r(t))$. Everything looks simple when considering the external dynamics only, but the internal dynamics can frequently be hard to predict. The orientation of the ship, given by the yaw angle psi, also needs to be analyzed. The controlled motion will not necessarily have the ship aligned with the tangent of the trajectory, for instance. Firstly, the ship control system that only focuses on the position variables (x, y) may equally well result in the ship moving backward along the line; a behavior that is not really desirable with respect to energy efficiency or for passenger comfort. Secondly,

there will always be environmental disturbances: currents, wind, and waves, and we need to make a thorough stability analysis of the internal dynamics in order to guarantee sufficient robustness properties for these. So to conclude, if you reduce the number of DOF that you seek to control, in order to achieve a fully actuated control problem, then you need to consider the internal dynamics very carefully when dealing with underactuated marine vessels.

If we follow the other approach to controlling underactuated marine vessels, where we seek to control more degrees of freedom than we have independent control inputs, then we not only have an underactuated marine vessel at hand, but we also have an underactuated control problem. This is a challenging control problem, and we will now see how this can be solved for path following and maneuvering control.

Path Following and Maneuvering Control of Underactuated Marine Vessels

For path following control systems, the control objective is to make the vessel follow a given path $\mathcal{P}$, often defined as a parametrized path

$$Y_d := \{y \in \mathbb{R}^m : \exists \theta \in \mathbb{R} \text{ such that } y = y_d(\theta)\}$$

where $m \leq n$ and y_d is continuously parametrized by the path variable θ. The control objective is thus to force the output y to converge to the desired path $y_d(\theta)$: $\lim_{t \to \infty} |y(t) - y_d(\theta(t))| = 0$. This constitutes a geometric task. When there is also a dynamic task, for instance, a speed assignment like forcing the path speed $\dot{\theta}$ to converge to a desired speed $v_s(\theta(t), t)$

$$\lim_{t \to \infty} |\dot{\theta}(t) - v_s(\theta(t), t)| = 0$$

then the control problem is an output maneuvering problem (Skjetne et al. 2004).

Line-of-sight (LOS) guidance control has proven to be a powerful tool for path following and maneuvering control of underactuated vessels. LOS guidance is much used in practice for manual control of ships, where the helmsman typically will steer the vessel toward a point lying a constant distance, called the look-ahead distance, ahead of the vessel along the desired path. LOS guidance is simple, intuitive, and easy to tune, and it can be shown that it provides nice path convergence properties (Breivik and Fossen 2004; Børhaug et al. 2008; Caharija et al. 2012; Fredriksen and Pettersen 2006; Lefeber et al. 2003). For the simplified case without any environmental disturbances and when the desired path is a straight line, the LOS guidance law for an underactuated surface vessel is given by

$$\psi_d = \psi_{\text{LOS}} = -\tan^{-1}\left(\frac{y}{\Delta}\right), \quad \Delta > 0$$

where y is the cross-track error. The angle ψ_{LOS} is called the line-of-sight (LOS) angle, and geometrically, it corresponds to the orientation of the vessel when headed toward the point that lies a distance $\Delta > 0$ ahead of the vessel along the path $y = 0$, cf. Fig. 1. The look-ahead distance Δ is a control design parameter.

In order to handle ocean currents and other environmental disturbances such as wind and waves, the LOS guidance law can be extended with integral action

$$\psi_{\text{LOS}}^m = -\tan^{-1}\left(\frac{y + \sigma y_{\text{int}}}{\Delta}\right), \quad \Delta > 0$$
$$\dot{y}_{\text{int}} = \frac{\Delta y}{(y + \sigma y_{\text{int}})^2 + \Delta^2}$$
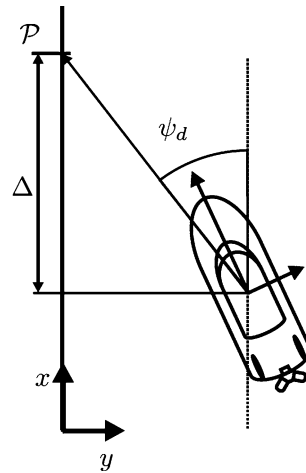
Underactuated Marine Control Systems, Fig. 1 Illustration of LOS guidance

where $\sigma > 0$ is a design parameter, an integral gain, and $\Delta > 0$ has the same interpretation as above. The integral effect will generate a sideslip angle that allows the vessel to stay on the desired path even though affected by environmental disturbances with components normal to the path, even though the vessel has no control forces to act in the sideways direction.

Various standard control techniques can readily be used to track the above guidance commands. LOS guidance can also be extended to the 3D case for path following/maneuvering control of underactuated autonomous underwater vehicles (AUV), cf. the references given above.

Summary and Future Directions

Underactuated marine vessels are vessels for which the dimension of the configuration space exceeds that of the control input space. There are two main approaches to design control systems for underactuated marine vessels. The first approach reduces the number of degrees of freedom (DOF) that it seeks to control, such that the number of DOF equals the number of independent control inputs. The control problem is then a fully actuated control problem, something that simplifies the control design problem significantly, but special attention then has to be given to the inherent internal dynamics that has to be carefully analyzed. The other approach to design control systems for underactuated marine vessels seeks to control all DOF using only the limited number of control inputs available. The control problem is then an underactuated control problem, and this is a quite challenging control problem. In this entry, it is shown how line-of-sight methods can solve the underactuated control problems that arise from path following and maneuvering control of underactuated marine vessels.

Future developments of underactuated marine control systems will include solving more underactuated control problems of marine vessels taking into account both the complete mathematical model of the vessels and also advanced mathematical models of all the environmental disturbances in both 2D and 3D.

Cross-References

▶ Mathematical Models of Ships and Underwater Vehicles
▶ Motion Planning for Marine Control Systems
▶ Underactuated Robots

Bibliography

Aguiar AP, Pascoal AM (2007) Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents. Int J Control 80:1092–1108

Breivik M, Fossen TI (2004) Path following of straight lines and circles for marine surface vessels. In: Proceedings of 6th IFAC conference on control applications in marine systems, Ancona, pp 65–70

Børhaug E, Pavlov A, Pettersen KY (2008) Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In: Proceedings of 47th IEEE conference on decision and control, Cancun, 9–11 Dec 2008, pp 4984–4991

Caharija W, Pettersen KY, Gravdahl JT, Børhaug E (2012) Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. In: Proceedings of 51th IEEE conference on decision and control, Maui, Dec 2012, pp 528–535

Encarnacao P, Pascoal AM, Arcak M (2000) Path following for marine vehicles in the presence of unknown currents. In: Proceedings of 6th IFAC international symposium on robot control, Vienna, 21–23 Sept 2000, pp 469–474

Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester/West Sussex

Fredriksen E, Pettersen KY (2006) Global K-exponential way-point maneuvering of ships: theory and experiments. Automatica 42:677–687

Goldstein H (1980) Classical mechanics, 2nd edn. Addison-Wesley, Reading

Healey AJ, Lienard D (1993) Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE J Ocean Eng 18:327–339

Indiveri G, Zizzari AA (2008) Kinematics motion control of an underactuated vehicle: a 3D solution with bounded control effort. In: Proceedings of 2nd IFAC workshop on navigation, guidance and control of underwater vehicles. Killaloe, Ireland

Isidori A (1995) Nonlinear control systems, 3rd edn. Springer, London

Lapierre L, Soetanto D, Pascoal AM (2003) Nonlinear path following with applications to the control of

autonomous underwater vehicles. In: Proceedings of 42nd IEEE conference on decision and control, Maui, Dec 2003, pp 1256–1261

Lefeber AAJ, Pettersen KY, Nijmeijer N (2003) Tracking control of an under-actuated ship. IEEE Trans Control Syst Technol 11:52–61

Nijmeijer H, van der Schaft AJ (1990) Nonlinear dynamical control systems. Springer, New York

Oriolo G, Nakamura Y (1991) Control of mechanical systems with second-order nonholonomic constraints: underactuated manipulators. In: Proceedings of 30th IEEE conference on decision and control, Brighton, Dec 1991, pp 2398–2403

Pettersen KY, Egeland O (1996) Exponential stabilization of an underactuated surface vessel. In: Proceedings of 35th IEEE conference on decision and control, Kobe, Dec 1996, pp 967–972

Pettersen KY, Egeland O (1999) Time-varying exponential stabilization of the position and attitude of an underactuated autonomous underwater vehicle. IEEE Trans Autom Control 44:112–115

Skjetne R, Fossen TI, Kokotivic PV (2004) Robust output maneuvering for a class of nonlinear systems. Automatica 40:373–383

Tarn T-J, Zhang M, Serrani A (2003) New integrability conditions for classifying holonomic and nonholonomic systems. In: Rantzer A, Byrnes CI (eds) Directions in mathematical systems theory and optimization, Springer, Berlin/Heidelberg, pp 317–331

Underactuated Robots

Kevin M. Lynch
Mechanical Engineering Department,
Northwestern University, Evanston, IL, USA

Abstract

Underactuated robots, robots with fewer actuators than degrees of freedom, are found in many robot applications. This entry classifies underactuated robots according to their dynamics and constraints and provides an overview of controllability, stabilization, and motion planning.

Keywords

Nonholonomic constraints; Nonlinear control; Underactuation

Introduction

An *underactuated robot* is a robot with fewer actuators (control inputs) than the number of variables describing its configuration (degrees of freedom). Some robots have this property unavoidably, while others are specifically designed this way, perhaps to save the cost of actuators. Examples include:

- **A cart and pendulum (inverted pendulum).** This system has two degrees of freedom, the linear position of the cart and the angle of the pendulum, but only one control input, the acceleration of the cart.

- **A car.** A car has only two control inputs (steering and forward/backward speed) but at least three degrees of freedom: the position (x, y) and orientation θ of the chassis. If the steering and/or rolling angles of the wheels are included in the representation of the configuration, the car has even more degrees of freedom.

- **A walking robot.** When a biped steps with one foot in the air and the toes of the other foot on the ground, there is no actuator at the toes to directly control the angle between the foot and the ground.

- **A quadrotor flying robot.** A quadrotor has four control currents driving the four propellers, but its configuration is described by six variables: (x, y, z) position and roll, pitch, and yaw.

- **An underactuated robot hand.** Robot hands generally have many joints, up to four per finger for anthropomorphic hands. To reduce cost, a small number of motors (as few as one) may be used to open and close the fingers, with joint motions coupled by springs.

- **Robot manipulation.** When a robot arm and hand manipulates a rigid object, the entire system, taken together, has at least six more degrees of freedom than actuators – the six degrees of freedom of the object.

In all underactuated robot systems of interest, the fewer control inputs are somehow coupled to all of the degrees of freedom. This entry focuses on coupling through the inertia matrix and kinematic constraints. In addition, this entry

U

focuses on control of the full configuration, or more generally the state, of the robot system. Other goals, such as successfully grasping an object with a compliant underactuated hand, are outside the scope of this entry.

Classification of Underactuated Robots

The robot has n degrees of freedom, and its configuration is written in local coordinates as a column vector $q \in \mathbb{R}^n$. If the robot is described as a *kinematic* system, then its state x is simply q and the control inputs are velocities. If the robot is a *mechanical* system, then its state is $x = [q^T, \dot{q}^T]^T$ and the control inputs are accelerations (forces). Let p denote the dimension of the state space, where $p = n$ for a kinematic system and $p = 2n$ for a mechanical system.

The equations of motion of an underactuated robot can be written in the control-affine form

$$\dot{x} = f(x) + \sum_{i=1}^{m} u_i g_i(x) \quad \text{where } m < n. \quad (1)$$

The vector field $f(x)$ is a *drift vector field* describing the unforced motion of the robot, the $g_i(x)$ are linearly independent *control vector fields* describing how the controls act on the robot, and $u = [u_1, \ldots, u_m]^T$ is the control. Kinematic systems are commonly *drift-free* ($f(x) = 0$). For a mechanical system, the drift field $f(x)$ typically includes velocities acting on positions and gravity acting on velocities.

The fact that the number of controls m is less than the number of degrees of freedom n can be viewed as $n - m$ constraints on the motion. For a kinematic system, these are velocity constraints. For a mechanical system, these are acceleration constraints. In addition, a mechanical system may be subject to a separate set of k velocity constraints, often called *Pfaffian* constraints, of the form

$$A(q)\dot{q} = 0, \quad (2)$$

where $A(q) \in \mathbb{R}^{n \times k}$. Such constraints arise from conservation laws and rolling without slip, for example.

Understanding the integrability of these constraints is key to understanding the controllability of underactuated robots (section "Determining Controllability"). For example, if acceleration constraints can be integrated to yield equivalent velocity constraints, then the dimension of the space of reachable velocities of the mechanical system is reduced. If velocity constraints can be integrated to yield equivalent configuration constraints, then the dimension of the reachable configuration space is reduced. If some velocity constraints are integrable to configuration constraints, we simply eliminate those configuration variables from the description of the system so we can focus on the controllable degrees of freedom. Velocity constraints that cannot be integrated are called *nonholonomic*, while configuration constraints are called *holonomic*.

Based on the type of constraints, we can classify underactuated robots into three categories – pure kinematic, pure mechanical, and mixed kinematic and mechanical – as described below.

Pure Kinematic

This category consists of systems with velocities as inputs, as well as mechanical systems that can be modeled by a *kinematic reduction* that has time-differentiable velocities as controls (Bullo and Lewis 2004; Bullo et al. 2002). (The actual acceleration controls of the original system are the time derivatives of these velocities.) Examples of mechanical systems that can be reduced to kinematic systems include systems with actuators for every degree of freedom (fully actuated systems, of little interest here) and mechanical systems whose acceleration constraints can be completely integrated to equivalent velocity constraints.

Example 1 (Upright rolling wheel) Consider a wheel of radius R rolling upright on a horizontal plane (Fig. 1a). The center of the wheel is (p_x, p_y, p_z), and the orientation is described by its "leaning" angle θ, rolling angle ψ, and heading angle ϕ. The constraints that the wheel
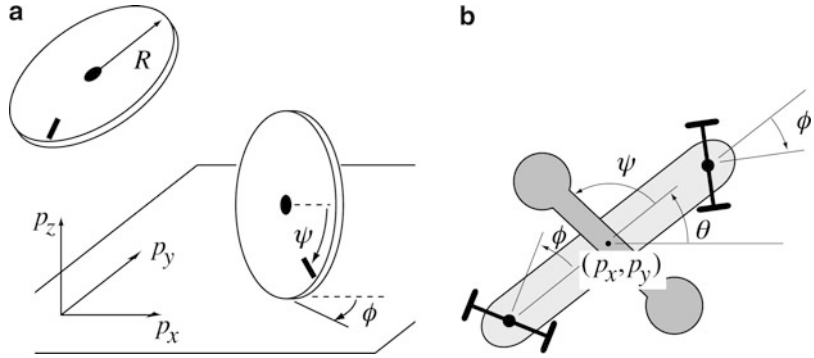
remain upright and touching the plane can be
written differentially as $\dot{p}_z = 0$ and $\dot{\theta} = 0$,
but these constraints can be integrated to the
equivalent configuration constraints $p_z = R$ and
$\theta = 0$, so we eliminate these variables from the
description of the configuration and focus on the
remaining four coordinates.

Writing the configuration vector as $q = [p_x, p_y, \psi, \phi]^T$ and the two control inputs as
the rolling velocity $u_1 = \dot{\psi}$ and the heading rate
of change $u_2 = \dot{\phi}$, the control system is

$$\dot{q} = u_1 g_1(q) + u_2 g_2(q),$$

where $g_1(q) = [R\cos\phi, R\sin\phi, 1, 0]^T$ and
$g_2(q) = [0, 0, 0, 1]^T$. Implicit in these equations
of motion are the two rolling constraints
$A(q)\dot{q} = 0$, where

$$A(q) = \begin{bmatrix} 1 & 0 & -R\cos\phi & 0 \\ 0 & 1 & -R\sin\phi & 0 \end{bmatrix}.$$

These velocity constraints cannot be integrated to
equivalent configuration constraints.

Example 2 (Reaction-wheel satellite) The three-
dimensional orientation of a satellite can be con-
trolled by spinning internal reaction wheels. The
controls to the reaction wheels are torques. By
conservation of angular momentum, the total an-
gular momentum P of the satellite is subject to
the constraints

$$P = J\omega + \sum_i J_i \omega_i = \text{constant},$$

where J is the inertia of the satellite body, ω is its
angular velocity, J_i is the inertia of momentum
wheel i, and ω_i is its angular velocity. These
constraints are velocity constraints – given the
angular velocity of the momentum wheels, the
angular velocity of the satellite is known. Thus,
we can treat the original mechanical system as
a kinematic system with (differentiable) angular
velocities of the momentum wheels as inputs.
If the system satisfies $P = 0$, the kinematic
reduction is drift-free.

While satellite orientation is commonly
controlled using three orthogonal reaction
wheels (a fully actuated system), two reaction
wheels suffice to control the orientation of
the kinematic reduction in the case $P = 0$.
This is apparent from the fact that successive
rotations about two orthogonal body-fixed axes
(e.g., body-referenced ZYZ Euler angles) are
sufficient to arbitrarily orient a rigid body in
space.

Pure Mechanical

This category consists of mechanical systems
without any velocity constraints.

Example 3 (3R robot arm with a passive joint)
The dynamics of a robot arm are determined by
its inertia matrix $M(q)$, from which the kinetic
energy $K = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ is derived, and its
potential energy $V(q)$. If one of the joints of the
arm rotates freely without an actuator, the arm
is underactuated. One such robot is a planar arm
with two actuated joints and one passive (Bullo
and Lynch 2001; Lynch et al. 2000). For this

robot, the acceleration constraint arising from the lack of an actuator cannot be integrated to an equivalent velocity constraint.

Mixed Kinematic and Mechanical

This category consists of mechanical systems with both (1) velocity constraints and (2) acceleration constraints that cannot be integrated.

Example 4 (Snakeboard) The snakeboard is a skateboard with steerable wheels. The rider can locomote without touching the ground by twisting his or her body while steering the wheels. The configuration of a robotic model of the snakeboard and rider (Ostrowski et al. 1994) consists of the position (x, y) and orientation θ of the board, the steering angle of the wheels (assumed to be coupled to be equal and opposite), and the angle of a reaction wheel representing the rider (Fig. 1b). The controls are the steering torque to the wheels and the driving torque of the reaction wheel. This system is mixed because of the presence of the no-slipping constraint at the wheels.

While in some cases it is obvious whether velocity or acceleration constraints can be integrated to equivalent constraints on configuration or velocity, respectively, in general this is not trivial. Instead of attempting to determine the integrability of constraints, we typically study the reachable sets of the system (1). This is the topic of controllability of nonlinear systems, section "Determining Controllability".

Underactuated robots can also be classified according to the set of available controls $\mathcal{U} \subseteq \mathbb{R}^m$. For example, the control set could be a discrete set of points in $\mathbb{R}^m$, or only nonnegative values, or a bounded set of $\mathbb{R}^m$ containing the origin in the interior. For simplicity, assume $u \in \mathcal{U} = \mathbb{R}^m$.

Control Challenges

Determining Controllability

For linear systems of the form $\dot{x} = Ax + Bu$, $x \in \mathbb{R}^p$, $u \in \mathbb{R}^m$, there is one notion of controllability, determined by the Kalman rank condition (KRC). If the rank of the matrix

$$[B \ \ AB \ \ A^2 B \ \ \dots \ \ A^{p-1}B]$$

is p, then it is possible to transfer the system from any state to any other state in finite time.

Most underactuated systems of the form (1), such as all of the examples given above, are non-linear systems, however. For nonlinear systems, there are many possible notions of controllability (see Bullo and Lewis 2004; Lynch et al. 2011; Nijmeijer and van der Schaft 1990; Sussmann 1983). Some examples include:

- *Small-time local accessibility (STLA) at x*: For any time $T > 0$, the reachable set starting from x at times $t < T$ contains a full-dimensional subset of the state space.
- *Small-time local controllability (STLC) at x*: For any time $T > 0$, the reachable set starting from x at times $t < T$ contains a neighborhood of x.
- *Global controllability*: The robot can reach any state from any other state.

STLC is strictly stronger than STLA. Neither implies global controllability nor does global controllability imply either of the local properties. STLA and STLC are illustrated in Fig. 2.

STLA can be tested by a Taylor expansion of flows along vector fields. A key object in this study is the *Lie bracket* of two vector fields $V_1(x)$ and $V_2(x)$, defined as the new vector field

$$[V_1, V_2] = \frac{\partial V_2}{\partial x} V_1 - \frac{\partial V_1}{\partial x} V_2.$$

If the system were to start from x and flow along V_1 for a short time ϵ, then V_2 for ϵ, then $-V_1$ for ϵ, then $-V_2$ for ϵ, a Taylor expansion shows that the net motion of the system would be $\epsilon^2 [V_1, V_2](x)$ (plus terms of order ϵ^3 and higher). If this direc-
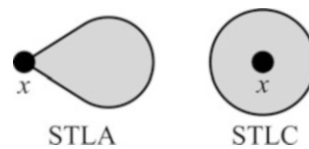
Underactuated Robots, Fig. 2 Example reachable sets in small time for systems that are STLA and STLC at x

tion is neither zero nor a linear combination of V_1 and V_2, then effectively a new motion direction has been created.

For the upright rolling wheel, the Lie bracket of g_1 (forward-backward rolling) and g_2 (turning) is

$$[g_1, g_2] = [R \sin \phi, -R \cos \phi, 0, 0]^T,$$

a sideways "parallel parking" motion. This new direction increases the dimension of the locally reachable set beyond what could be reached by a local linearization of the nonlinear system.

Roughly speaking, the *Lie algebra* of a set of vector fields $\mathcal{V}$ is the set of vector fields $\mathcal{V}$, all iterated Lie brackets of these vector fields, and their linear combination. For example, the Lie algebra of $\mathcal{V} = \{g_1, g_2\}$ includes $[g_1, g_2]$, $[g_1, [g_1, g_2]]$, $[g_2, [g_1, [g_1, g_2]]]$, etc., as well as their linear combinations. Deeper Lie brackets correspond to higher-order terms in the Taylor expansion of flows.

With these concepts, a theorem due to Chow (1939) says that a system (1) satisfies STLA at x if the dimension of the Lie algebra of $\{f, g_1, \ldots, g_m\}$ at x is p, the dimension of the state space. This is known as the *Lie algebra rank condition* (LARC). Most underactuated systems of interest satisfy the LARC but not the KRC. For the upright rolling wheel, the linearization at any q fails the KRC, but the four-dimensional configuration space is spanned by g_1, g_2, $[g_1, g_2]$, and $[g_2, [g_1, g_2]]$ at all q, satisfying the LARC. Therefore the system is STLA at all points.

The STLA property can be strengthened to STLC if the system additionally satisfies certain symmetry properties, allowing it to proceed both forward and backward along Lie bracket directions. For example, if $f(x) = 0$ and the control set $\mathcal{U}$ contains the origin in the interior, the LARC implies STLC. This is the case for the upright rolling wheel. More general notions of symmetry have also been derived (e.g., Sussmann 1987).

For mechanical systems, STLC can only hold at zero-velocity states where $f(x) = 0$. In addition, velocity constraints may prevent the system from reaching a $2n$-dimensional set in state space. A more relevant question may be whether the configuration alone can be locally controlled at zero-velocity states. Specialized Lie-algebraic controllability tests have been developed for configuration controllability of mechanical and mixed systems (Bullo and Lewis 2004; Bullo and Lynch 2001; Bullo et al. 2002; Lewis 2000).

Global controllability results often derive from STLC at all states for drift-free systems or from STLA and global properties of the vector fields or the topology of the state space (Choset et al. 2005).

Feedback Stabilization

For some underactuated robots, the linearization at a state x may satisfy the KRC. An example is an inverted pendulum linearized at a balanced equilibrium state. In this case, it is possible to derive a linear feedback controller, based on the linearization, to stabilize the balanced state.

For many underactuated systems of interest, however, the linearization at a desired state is not controllable. For such systems, a famous theorem due to Brockett (1983), plus subsequent strengthening, implies the following:

Theorem 1 *For any drift-free underactuated kinematic system of the form* (1)*, there exists no time-invariant continuous state feedback law that stabilizes the origin.*

For example, there exists no continuous state-feedback control law that can stabilize the upright rolling wheel to a desired configuration.

This obstruction to stabilizability has resulted in a number of different approaches to feedback control of underactuated systems, including (1) time-varying feedback control laws, (2) feedback control laws that are discontinuous in the state, and (3) two-degree-of-freedom controllers consisting of a motion planner plus a feedback controller for the easier problem of stabilizing the nominal trajectory. Strategies for planning nominal motions for two-degree-of-freedom controllers are discussed next.

Motion Planning

Given an initial state $x(0) = x_{\text{start}}$ and a goal state x_{goal}, the motion planning problem for a

system $\dot{x} = h(x, u)$ is to find a control history $u : [0, T] \rightarrow \mathcal{U}$ such that

$$x_{\text{goal}} = x_0 + \int_0^T h(x(s), u(s)) ds$$

while avoiding any obstacles that may be present in the environment. It may also be desired to minimize some notion of cost,

$$J = \int_0^T L(x(s), u(s)) ds.$$

One choice of $L(s)$ is $u^T(s)u(s)$, the square of the control effort.

Ideally the motion planning method would be *complete* (guaranteed to find a solution in finite time if one exists) or *probabilistically complete* (if a solution exists, the probability of finding a solution goes to one as time goes to infinity).

A variety of approaches to motion planning have been proposed in the robotics literature. Approaches that apply to underactuated systems include:

- *Search-based methods.* A popular class of search-based methods are rapidly exploring random trees (RRTs) and variants (LaValle and Kuffner 2001). These approaches offer probabilistic completeness for many systems, including systems with obstacles, but naïve implementations may be slow to find solutions, and the solutions generally do not satisfy optimality criteria.

- *Numerical optimization.* The control history can be converted to a finite parameterization using representations such as polynomials, cubic B-splines, wavelets, and truncated Fourier series. Numerical optimization methods can then be applied to solve the two-point boundary value problem while minimizing a cost function. Gradient-based numerical optimization methods may yield locally optimal solutions, but they may suffer from numerical convergence problems, and they may get stuck in local minima depending on an initial guess. Optimization methods that do not use gradient information potentially offer globally optimal

solutions, but typically at the expense of significantly longer computation times.

- *Fictitious input methods.* These methods assume that there is a direct control input available for each Lie bracket motion direction. These fictitious inputs are then converted to a sequence of feasible inputs utilizing the Campbell-Baker-Hausdorff-Dynkin expansion of flows (Lafferriere and Sussmann 1991). In general, these methods require iterative application to account for errors in the approximate conversion.

- *Trajectory transformation methods.* One way to deal with obstacles is to first use a global motion planner that is complete under the assumption that the robot has no motion constraints. Then the template unconstrained solution is iteratively subdivided into smaller pieces, with each piece replaced by an obstacle-free feasible trajectory generated by a local planner. If the system is drift-free and STLC at all configurations, then it is possible to develop a local planner that guarantees success of the transformation from an unconstrained trajectory to a feasible trajectory as the subdivisions get small enough (Laumond et al. 1994).

Often it is possible to exploit structure of the equations of motion beyond the general form (1). Making use of extra structure can reduce the computational complexity of motion planning.

- *Chained form, sinusoidal controls, and averaging.* Certain drift-free kinematic systems can be transformed to a canonical *chained form*. For systems in such a form, sinusoidal controls of integrally related frequencies can be chosen to drive one of the configuration variables to its desired value while having zero net effect on configuration variables already at their desired value. In this way, configuration variables can be driven sequentially to their desired values (Murray and Sastry 1993).

For many underactuated systems, sinusoidal controls can be used to achieve approximate motion in each Lie bracket direction needed to complete the LARC. The resulting periodic motions are sometimes called *gaits*, and motion planning can be

achieved using a finite set of gaits (Bullo and Lewis 2004; Ostrowski et al. 1994).

- *Differentially flat systems.* For certain underactuated systems with $u \in \mathbb{R}^m$, there exist a set of m functions w_i of the state, the control, and its derivatives,

$$w_i(x, u, \dot{u}, \ddot{u}, \ldots, u^{(r)}), \quad i = 1 \ldots m,$$

such that the states and control inputs can be expressed as functions of w and its time derivatives. The w_i are called *flat outputs*. The motion planning problem is to find $w(t), t \in [0, T]$, such that $w(0), \dot{w}(0), \ddot{w}(0), \ldots$ and $w(T), \dot{w}(T), \ddot{w}(T), \ldots$ satisfy the constraints specified by x_{start} and x_{goal}. The problem changes from constrained motion planning in the p-dimensional state space to finding a curve satisfying start and end constraints on w and its derivatives (Fliess et al. 1995; Sira-Ramirez and Agrawal 2004).

- *Kinematic reductions.* Motion planning in configuration space is a lower-dimensional problem than motion planning in configuration-velocity space. Therefore, when a mechanical system can be reduced to a kinematic equivalent, motion planning can be more efficient. Examples include mechanical systems that can be fully reduced to a kinematic system (like the reaction-wheel satellite) and mechanical systems that admit rank-1 kinematic reductions – vector fields on configuration space that can be followed at any speed, despite the underactuation constraints. These vector fields become primitives for motion planning on configuration space (Bullo and Lewis 2004; Bullo and Lynch 2001; Bullo et al. 2002; Choset et al. 2005).

Summary and Future Directions

Underactuated systems arise in all areas of robotics, including robot manipulation and aerial, ground, and underwater locomotion. Underactuation raises a number of challenging issues in robot motion planning and control.

While significant progress has been made, further research is needed on computationally efficient motion planning and robust stabilization of nominal trajectories. In addition, although this entry focuses on systems that can be described by a single set of dynamics, many interesting underactuated systems are hybrid systems that experience changing contact constraints. Examples include biped robots striding from one foot to the next and robot manipulators that manipulate objects with changing contact modes (grasping, rolling, pushing, etc.). Further work is needed to incorporate contact models, beyond simple kinematic constraints, and changing equations of motion in motion planning and control of hybrid underactuated systems.

Cross-References

- ▶ Controllability and Observability
- ▶ Differential Geometric Methods in Nonlinear Control
- ▶ Feedback Linearization of Nonlinear Systems
- ▶ Feedback Stabilization of Nonlinear Systems
- ▶ Hybrid Dynamical Systems, Feedback Control of
- ▶ Lie Algebraic Methods in Nonlinear Control
- ▶ Nonlinear Zero Dynamics
- ▶ Underactuated Marine Control Systems
- ▶ Walking Robots
- ▶ Wheeled Robots

Recommended Reading

Introductions to underactuated robot systems can be found in Choset et al. (2005), Lynch et al. (2011), and Murray et al. (1994).

While this entry focuses on configuration spaces modeled locally as $\mathbb{R}^n$, most robotic systems consist of rigid bodies whose positions and orientations can be described globally as elements of the Lie group $SE(3)$ or one of its subgroups: $SE(2)$, $SO(3)$, or $SO(2)$. Geometric methods for control of underactuated systems make use of the extra structure of Lie groups and their Lie algebras, symmetries, and concepts

from geometric mechanics such as tangent and cotangent bundles, Riemannian metrics on manifolds, symplectic manifolds, connections, fiber bundles, covariant derivatives, etc. Excellent treatments can be found in Bloch et al. (2003), Bullo and Lewis (2004), and Murray et al. (1994).

Bibliography

Bloch AM, Baillieul J, Brouch PE, Marsden JE (2003) Nonholonomic mechanics and control. Springer, New York

Brockett RW (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston

Bullo F, Lewis AD (2004) Geometric control of mechanical systems. Springer, New York/Heidelberg/Berlin

Bullo F, Lynch KM (2001) Kinematic controllability for decoupled trajectory planning of underactuated mechanical systems. IEEE Trans Robot Autom 17(4):402–412

Bullo F, Lewis AD, Lynch KM (2002) Controllable kinematic reductions for mechanical systems: concepts, computational tools, and examples. In: International symposium on the mathematical theory of networks and systems, South Bend, IN, Aug 2002

Choset H, Lynch KM, Hutchinson S, Kantor G, Burgard W, Kavraki L, Thrun S (2005) Principles of robot motion. MIT, Cambridge

Chow W-L (1939) Uber systemen von linearen partiellen differentialgleichungen erster ordnung. Math Ann 117:98–105

Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. Int J Control 61(6):1327–1361

Lafferriere G, Sussmann H (1991) Motion planning for controllable systems without drift. In: IEEE international conference on robotics and automation, Sacramento, pp 1148–1153

Laumond J-P, Jacobs PE, Taïx M, Murray RM (1994) A motion planner for nonholonomic mobile robots. IEEE Trans Robot Autom 10(5):577–593

LaValle SM, Kuffner JJ (2001) Rapidly-exploring random trees: progress and prospects. In: Donald BR, Lynch KM, Rus D (eds) Algorithmic and computational robotics: new directions. A. K. Peters, Natick

Lewis AD (2000) Simple mechanical control systems with constraints. IEEE Trans Autom Control 45(8):1420–1436

Lynch KM, Shiroma N, Arai H, Tanie K (2000) Collision-free trajectory planning for a 3-DOF robot with a passive joint. Int J Robot Res 19(12): 1171–1184

Lynch KM, Bloch AM, Drakunov SV, Reyhanoglu M, Zenkov D (2011) Control of nonholonomic and underactuated systems. In: Levine W (ed) The control systems handbook: control system advanced methods, 2nd edn. Taylor and Francis, Boca Raton

Murray RM, Sastry SS (1993) Nonholonomic motion planning: steering using sinusoids. IEEE Trans Autom Control 38(5):700–716

Murray RM, Li Z, Sastry SS (1994) A mathematical introduction to robotic manipulation. CRC Press, Boca Raton

Nijmeijer H, van der Schaft AJ (1990) Nonlinear dynamical control systems. Springer, New York

Ostrowski J, Lewis A, Murray R, Burdick J (1994) Nonholonomic mechanics and locomotion: the snakeboard example. In: IEEE international conference on robotics and automation, San Diego, CA, pp 2391–2397

Sira-Ramirez H, Agrawal SK (2004) Differentially flat systems. CRC Press, New York

Sussmann HJ (1983) Lie brackets, real analyticity and geometric control. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston

Sussmann HJ (1987) A general theorem on local controllability. SIAM J Control Optim 25(1):158–194

Validation and Verification Techniques and Tools

Christine M. Belcastro
NASA Langley Research Center, Hampton,
VA, USA

Abstract

Validation and verification (V&V) of advanced control systems is required for their use in fielded systems. A comprehensive V&V process involving analysis, simulation, and experimental testing should be used to assess closed-loop system performance and identify system limitations. This entry discusses current V&V methods and tools as well as future research directions for safety-critical control applications.

Keywords

Closed-loop system stability and performance; Software Verification; Stability and performance robustness; Uncertainties and uncertainty models; Validation of Safety-Critical Systems

Introduction

Control system validation and verification (V&V) is an assurance that the closed-loop system (i.e., the control system acting on the plant being controlled) remains stable and performs within acceptable performance metrics across the operational region of application. Basic definitions of validation and verification are given below (IEEE 2011).

Validation: The assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers.

Verification: The evaluation of whether or not a product, service, or system complies with a regulation, requirement, specification, or imposed condition. It is often an internal process.

Control system validation can therefore be thought of as a confirmation that the algorithms are performing their intended functions and an affirmation of their effectiveness in performing these functions. Validation is a rigorous evaluation process that should involve clearly identifying system limitations, including regions of operation within which stability or acceptable levels of performance are not guaranteed. Verification can be thought of as a confirmation that the system implementation in software and hardware is correctly executing the algorithms as designed (and validated). This includes a rigorous evaluation of system requirements and specifications and a clear determination of whether or not they have been met. V&V methods include analysis, simulation, and experimental testing, which are (ideally) applied in an integrated or iterative manner to corroborate results across methods of

evaluation. In the case of aircraft flight control and other safety-critical control applications, the V&V process must ultimately lead to system certification. The following subsections summarize control system V&V analytical, simulation, and experimental test methods in terms of the current (or recommended) state of practice. A summary and some future research directions for safety-critical control applications are also provided.

Control System Validation

Validation methods, involving analysis, simulation, and experimental testing, are utilized to ensure against errors and significant deficiencies in the underlying control system algorithms under realistic operational conditions for the intended application. System weaknesses and limitations are also identified through the validation process. Control system validation begins with an analysis of closed-loop system stability, performance, and robustness. Linear systems theory forms the basis for analytical stability and robustness methods and the associated software tools that are currently available for closed-loop system validation. In current practice, stability of nonlinear systems is determined by evaluating the stability of the linearized closed-loop system at numerous equilibrium points across the operating range (or envelope) of the system. Closed-loop stability is assessed by computing the eigenvalues of the linearized closed-loop system at a number of equilibrium points in the region of operation. It should be noted, however, that stability is not guaranteed between the operating points analyzed. Moreover, if the control system utilizes gain scheduling across the operating envelope, stability cannot be guaranteed for interpolated gains between the design points. Relative stability is determined by gain and phase margins for single-input, single-output (SISO) systems and by the multivariable stability margin (see, e.g., Lavretsky and Wise 2013) for multiple-input, multiple-output (MIMO, or multivariable) systems. Time-delay margins, defined as the minimum time delay required to destabilize the closed-loop system, can be determined from

phase margin and verified in nonlinear simulation. Stability robustness is assessed in terms of uncertainties, including parametric uncertainties and unmodeled dynamics in the mathematical model of the plant. Advanced robustness analysis methods based on the structured singular value (Zhou et al. 1996) require the uncertainties to be separated from the nominal plant into what is termed a linear fractional transformation (LFT). Advanced robustness methods can be used to assess stability and performance robustness, as well as worst-case combinations of uncertainties that result in destabilization, loss of performance, or the lowest robustness margins. LFT models can be formulated for the analysis of nonlinearities, expressed as multivariate polynomials, around a trim condition or over subregions within the operational envelope. Stability over a region or subregion of the operating envelope can be guaranteed using linear parameter-varying (LPV) methods (see Apkarian and Gahinet 1995; Packard 1994; Rugh and Shamma 2000; Wu et al. 1996). Nonlinear stability and control are addressed more fully in Slotine and Li (1991) and Khalil (2002), as well as in numerous other texts. Analytical methods and software tools are available in Matlab®, using the Control System Toolbox™ and Robust Control Toolbox™. LFR/LFT modeling methods and software tools are described in Magni (2004, 2006), Hecker et al. (2005), Varga et al. (1998), and Belcastro et al. (2005) and provided in the Robust Control Toolbox®. An LPV Toolbox™ is also under development and will become available soon (see Balas et al. 2013b).

Performance is usually assessed using a high-fidelity simulation of the plant under expected operational conditions. The simulation should include nonlinear plant dynamics, noise, disturbances, and any other phenomena anticipated under operation. Nominal performance is assessed in terms of the control system design objectives, which typically include (at a minimum) closed-loop steady-state tracking and transient response characteristics. Transient response characteristics typically include delay time, rise time, peak time, maximum overshoot, and settling time (see, e.g., Ogata 1970). Steady-state tracking error is also

typically assessed. Performance robustness can be evaluated using Monte Carlo simulation techniques (see, e.g., Kroese et al. 2011), in which parameters and operational conditions are varied over numerous simulation runs in order to statistically evaluate an extensive set of uncertainties and operational variations. Stability robustness can also be assessed using Monte Carlo simulation techniques and by utilizing worst-case uncertainties and time-delay margins obtained during analysis. If the plant is a vehicle or robotic manipulator to be operated by a human, the handling qualities must also be evaluated to assess human-system interfaces and interactions. A real-time high-fidelity simulation with a human interface representative of the operational environment is required for this evaluation. For aircraft, piloted simulation evaluations are conducted using a cockpit mock-up, and handling qualities are assessed under various scenarios using the Cooper-Harper Scale (Cooper and Harper 1969). Susceptibility to operator-induced oscillations, for example, resulting from time delays in the controlled response, may typically be uncovered using operator-in-the-loop simulation evaluations.

Experimental testing should be conducted under realistic conditions that cover the entire (potential) operational space of the plant being controlled in order to assess realistic operational performance. For aircraft, this includes flight testing using full-scale and/or subscale test vehicles under nominal and off-nominal conditions. If the analytical and simulation evaluations provide a good match to the experimental evaluations, the test matrix can be comprised of key high-risk conditions to confirm desired behavior.

Validation methods should be applied in an iterative manner comparing results from the analysis, simulation, and experimental tests and going back to reevaluate in one domain based on results from another. For example, analysis results should provide good predictions of results seen in simulation and experimental testing. If a good match is not obtained, the analysis model may have to be improved or another analysis method

utilized to reduce conservatism in the result. Similarly, simulation results should provide a good prediction of experimental test results.

Control System Verification

System verification is ideally performed by or in collaboration with a computer science specialist to ensure against errors in the software/hardware implementation of the control algorithms. Control system verification begins with an analysis (Rushby 1995, 2009) of the software and hardware implementation requirements to ensure completeness and accuracy in the system specification. Several refinement steps are taken to transform the control system requirements to those implementable on the actual avionics hardware to be fielded. Verification, consisting of tests and analyses, is required to confirm requirements traceability and compliance from one refinement to another, to confirm accuracy of the algorithms, and to assure compliance and robustness of the final code with respect to the original control algorithms, to ensure that no errors are introduced from the refinement itself or related to the target computing platform. Formal analysis methods can be used to evaluate software logic and other software mechanisms for correctness under all operational conditions and to provide correctness proofs. Formal methods can also be utilized for model checking to verify system properties through an exhaustive search of all possible states that can be entered during execution (Berard et al. 1998). One software verification tool is called PVS (Owre et al. 1992), developed by SRI International, and is available on the Internet as an open-source software tool. Other methods and tools are also available from SRI International, as well as other sources.

Simulation techniques are used to evaluate software code, modules, subsystems, and the full system. Once the software has been verified, it is implemented on actual or representative hardware and evaluated using hardware-in-the-loop simulation and experimental testing. Experimental testing should include laboratory evaluations under all possible operational conditions and in

a relevant application environment. For aircraft, this would include flight testing of the control system on the actual avionics hardware to be fielded.

Summary and Future Directions

This entry has summarized current and recommended practices for control system V&V, including methods and tools for analysis, simulation, and experimental testing. The V&V process ensures against errors and deficiencies in the underlying system algorithms (validation) and in its software/hardware implementation (verification). Analysis, simulation, and experimental testing are performed iteratively to utilize and confirm results between evaluation techniques.

A comprehensive validation process is performed to assure control system effectiveness across the operational envelope of the plant and to identify system limitations and weaknesses. Current analysis methods for control system validation are typically based on linear systems theory and focus on nominal operations under model uncertainties and anticipated disturbances (e.g., noisy measurement signals). This analysis includes stability, performance (e.g., tracking accuracy), and robustness. Advanced robust control analysis methods have been applied to safety-critical applications, such as aircraft (see Fielding et al. 2002; Varga et al. 2012), to assess robust stability and performance. These two references provide a global optimization-based worst-case approach as a "necessary condition" technique for flight control system validation or as a "sufficient condition" technique for invalidation. High-fidelity nonlinear simulation evaluations are performed in batch and real time to assess robustness under system and operational uncertainties and to assess interface effectiveness for human-in-the-loop operations (if applicable). Experimental testing under realistic operationally relevant conditions is performed across key operating conditions to confirm analytical and simulation predictions.

System verification is performed to ensure correctness of the hardware/software implementation. Various analysis and testing methods, including advanced formal analysis methods, are used to assess completeness of the system requirements and specification and software elements (e.g., logic). Model checking techniques are used to verify system properties. Code is tested in simulation and on representative or actual hardware under realistic operationally relevant conditions.

Future research directions will enable the V&V of nonlinear and adaptive control systems (see, e.g., Hovakimyan and Cao 2010; Tallant et al. 2004) that improve performance under highly uncertain conditions, as well as the V&V of complex integrated safety-critical systems for operation under off-nominal and hazardous conditions (see Belcastro 2010, 2012). These systems will include diagnostic and prognostic algorithms for integrated vehicle health management, resilient control systems that enable the detection and mitigation of multiple hazards, supervisory systems that provide safety assurance (for safety-critical operations), and intelligent interface and decision-based systems that enable human-optional and fully autonomous operations. These systems will inherently involve stochastic decision-making and nonlinear and adaptive control algorithms. V&V of these future systems poses significant technical challenges and is the subject of current research. Some of these challenges include the following: (1) development and validation of multidisciplinary simulation models for characterizing hazardous condition effects; (2) validation of adaptive, diagnostic/prognostic, and reasoning algorithms under numerous off-nominal and hazardous conditions; (3) verification of software-intensive highly complex systems; and (4) determining a level of confidence in V&V results for hazardous application domains that cannot be fully replicated during the evaluations.

Research on modeling and simulation methods is being performed to characterize multidisciplinary effects of off-nominal and hazardous conditions, and validation of these models can be difficult. For aircraft applications, hazardous

conditions relate to aircraft loss of control (LOC) and precursor conditions. LOC is a complex and highly nonlinear phenomenon for which there is little available data. Hazardous conditions considered in this research include vehicle upset conditions (e.g., stall/departure), vehicle impairment conditions (e.g., failure, icing, and damage), and external disturbances (e.g., inclement weather and wake vortices). Multidisciplinary models under development include aerodynamic, propulsion, and airframe structure effects. For example, simulation models for characterizing aircraft flight dynamics and control effects under upset conditions are currently being developed (see, e.g., Foster et al. 2005; Groen et al. 2012), as well as propulsion effects resulting from the associated reduced flow conditions (Liu et al. 2013). Model validation is being performed using available flight and accident data, as well as experimental testing in the laboratory and through subscale and full-scale flight testing. The enhanced high-fidelity simulation models resulting from this research will be used in the development and validation of onboard control systems designed to detect and mitigate these hazards.

Current research efforts for validating the above future systems include nonlinear robustness analysis methods and software tools (see (Chakraborty et al. 2011a, b), Balas et al. 2013a; Packard et al. 2010; Summers et al. 2013), nonlinear analysis methods for controlled systems (Gill et al. 2012; Kwatny et al. 2013), uncertainty quantification and robustness analysis methods for mixed uncertainties and multiple objectives (Kenny et al. 2012), and the analysis of stochastic filters (see, e.g., Reif et al. 1999; Rhudy et al. 2013a, b, c). The term "mixed uncertainty" refers to aleatory and epistemic uncertainties. Aleatory uncertainties are typically stochastic (or statistical) and represent operational or environmental uncertainties (e.g., turbulence) that cannot be altered or controlled during experiments or fielded applications. Epistemic uncertainties are typically deterministic and arise from lack of knowledge about the plant resulting from modeling assumptions, neglected effects (e.g., unmodeled dynamics), and parametric

uncertainties resulting from inaccurate measurements or operational variability. These analysis methods and tools will be used iteratively with simulation evaluations and experimental testing methods, as described herein, to comprehensively assess nonlinear and adaptive control systems that enable resilience under multiple hazards.

Current research efforts on software verification focus on argument-based safety assurance for highly complex integrated systems of systems; assessment tools for evaluating the safety and coordination of authority and autonomy assignments; methods for ensuring safety-critical properties of distributed systems; and the development of tools and techniques for assessing software-intensive systems in meeting performance safety objectives. Research on software-intensive systems includes the development of methods and tools to detect, diagnose, and predict adverse events due to a software fault or failure once the software has been verified and is in operation. Some recent references on this work include Holloway (2012), Xu et al. (2013), Driscol et al. (2012), Person et al. (2011), and Latorella and Feary (2011).

Research has been initiated on developing methodologies for determining (i.e., quantifying) the predictive capability of the validation process for systems designed to operate under conditions that cannot be fully replicated during evaluations. Predictive capability assessment is an evaluation of the validity and level of confidence that can be placed in the validation process and results under nominal and hazardous conditions (and their associated boundaries). The need for this evaluation arises from the inability to fully evaluate these technologies under actual hazards to be encountered by the fielded system. A detailed disclosure is required of model, simulation, and emulation validity for the off-nominal conditions being considered in the validation, interactions that have been neglected, assumptions that have been made, and uncertainties associated with the models and data. Cross-correlations should be utilized between analytical, simulation, and ground test and flight test results in order to corroborate the results and promote efficiency in covering the very

V

large space of operational and off-nominal and hazardous conditions being evaluated. The level of confidence in the validation process and results must be established for subsystem technologies as well as the fully integrated system. This includes an evaluation of error propagation effects across subsystems and an evaluation of integrated system effectiveness in mitigating hazardous conditions and preventing cascading errors, faults, and failures across subsystems. Metrics for performing this evaluation are also needed.

Cross-References

▶ Computer-Aided Control Systems Design: Introduction and Historical Overview
▶ Interactive Environments and Software Tools for CACSD
▶ Robust Synthesis and Robustness Analysis Techniques and Tools

Bibliography

Apkarian P, Gahinet P (1995) A convex characterization of gain-scheduled Hinf controllers. IEEE Trans Autom Control AC-40(5):853–864

Balas G, Chiang R, Packard A, Safonov M, Robust control toolbox™, Matlab® product family. The Mathworks Inc., Natick, MA, 1994–2014

Balas G, Packard A, Seiler P, Topcu U (2013a) Robustness analysis of nonlinear systems. University of Minnesota. Website http://www.aem.umn.edu/~AerospaceControl/

Balas GJ, Chaing R, Packard AK, Safonov M (2013b) Robust control toolbox. The Mathworks Inc., Natick, MA

Belcastro CM, Khong TH, Shin J-Y, Balas GJ, Kwatny HG, Chang B-C (2005) Uncertainty modeling for robustness analysis of control upset prevention and recovery systems. In: AIAA guidance, navigation, and control conference and exhibit, AIAA-2005-6427, San Francisco

Belcastro CM (2010) Validation and verification of future integrated safety-critical systems operating under off-nominal conditions. In: AIAA guidance, navigation and control conference, Toronto, Aug 2010

Belcastro CM (2012) Validation of safety-critical systems for aircraft loss-of-control prevention and recovery. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012

Berard B, Bidoit M, Finkel A, Laroussinie F, Petit A, Petrucci L, Schnoebelen P (1998) Systems and software verification: model-checking techniques and tools. Springer, Berlin

Chakraborty A, Seiler P, Balas GJ (2011a) Susceptibility of F/A-18 flight controllers to the falling-leaf mode: linear analysis. J Guid Control Dyn 34(1):57–72

Chakraborty A, Seiler P, Balas GJ (2011b) Susceptibility of F/A-18 flight controllers to the falling-leaf mode: nonlinear analysis. J Guid Control Dyn 34(1):73–85

Chen C-T (1998) Linear system theory and design, 3rd edn. Oxford University Press, New York

Control System Toolbox™, Matlab® product family. The Mathworks Inc., Natick, MA

Cooper GE, Harper RP Jr (1969) The use of pilot rating in the evaluation of aircraft handling qualities. AGARD report 567, Apr 1969

Doyle JC, Francis BA, Tannenbaum AR (1992) Feedback control theory, Macmillan, New York

Driscol K, Madl G, Hall B (2012) Modeling and analysis of mixed synchronous/asynchronous systems. NASA/CR-2012-217756, Sept 2012

Fielding C, Varga A, Bennani S, Selier M (eds) (2002) Advanced techniques for clearance of flight control laws. Springer, Berlin

Foster JV, Cunningham K, Fremaux CM, Shah GH, Stewart EC, Rivers RA, Wilborn JE, Gato W (2005) Dynamics modeling and simulation of large transport airplanes in upset conditions. In: AIAA guidance, navigation, and control conference, San Francisco

Gill SJ, Lowenberg MH, Krauskopf B, Puyou G, Coetzee E (2012) Bifurcation analysis of the NASA GTM with a view to upset recovery. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012

Groen E, Ledegang W, Field J, Smaili H, Roza M, Fucke L, Nooij S, Goman M, Mayrhofer M, Zaichik L, Grigoryev M, Biryukov V (2012) SUPRA – enhanced upset recovery simulation. In: AIAA guidance, navigation, and control conference, Minneapolis, Aug 2012

Hartmann AK (2009) Practical guide to computer simulations. World Scientific, Hackensack, New Jersey

Hecker S, Varga A, Magni J (2005) Enhanced LFR toolbox for Matlab. Aerosp Sci Technol 9(2):173–180

Holloway CM (2012) Towards understanding the DO-178C/ED-12C assurance case. In: Proceedings of the IET 7th international conference on system safety, Edinburgh, Oct 2012

Hovakimyan N, Cao C (2010) L1 adaptive control theory. Society for Industrial and Applied Mathematics, Philadelphia

IEEE (2011) IEEE guide–adoption of the Project Management Institute (PMI®) standard a guide to the Project Management Body of Knowledge (PMBOK® Guide), 4th edn. IEEE, p 452. doi:10.1109/IEEESTD.2011.6086685. Retrieved 7 Dec 2012

Kenny SP, Crespo LG, Giesy DP (2012) UQ tools: the uncertainty quantification toolbox – introduction and tutorial. NASA TM-2012-217561, Apr 2012

Khalil HK (2002) Nonlinear systems, 3rd edn. Prentice Hall, Upper Saddle River, New Jersey

Kwatny HG, Dongmo J-ET, Chang B-C, Bajpai G, Yasar M, Belcastro C (2013) Nonlinear analysis of aircraft loss of control. J Guid Control Dyn 36(1):149–162

Kroese DP, Taimre T, Botev ZI (2011) Handbook of Monte Carlo methods. Wiley series in probability and statistics. Wiley, New York

Latorella KA, Feary M (2011) NASA aviation safety programs: human factors focused work. In: 16th International symposium on aviation psychology, Dayton, 2–5 May 2011

Lavretsky E, Wise KA (2013) Robust and adaptive control. Springer, London

Liu Y, Claus RW, Litt JS, Guo T-H (2013) Simulating Effects of High Angle of Attack on Turbofan Engine Performance. In: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, Texas, 7–10 January 2013

Magni J-F (2004) Linear fractional representation toolbox – modeling, order reduction, and gain scheduling. ONERA technical report TR 6/08162 DSCD, Systems Control and Flight Dynamics Department, ONERA, July 2004

Magni J-F (2006) User manual of the linear fractional representation toolbox (version 2.0). Technical report 5/10403.01F, ONERA/DCSD. http://www.onera.fr/staff-en/jean-marc-biannic/docs/lfrtv20s.zip

Matlab®, The Mathworks Inc., Natick, MA

Ogata K (1970) Modern control engineering. Prentice-Hall, Englewood Cliffs

Owre S, Shankar N, Rushby J (1992) PVS: a prototype verification system. In: CADE 11, Saratoga Springs, June 1992

Packard AK (1994) Gain-scheduling via linear fractional transformations. Syst Control Lett 22:79–92

Packard A, Topcu U, Seiler P, Balas G (2010) Help on SOS. IEEE Control Syst Mag 30(4):18–23

Person SJ, Yang G, Rungta N, Khurshid S (2011) Directed incremental symbolic execution. In: 32nd ACM SIGPLAN conference on programming design and implementation, San Jose, June 4–8 2011

Reif K, Gunther S, Yaz E, Unbehauen R (1999) Stochastic stability of the discrete-time extended Kalman filter. IEEE Trans Autom Control 44(4):714, 728

Rhudy M, Gu Y, Napolitano MR (2013a) An analytical approach for comparing linearization methods in EKF and UKF. Int Journal of Adv Robot Syst, 2013, Vol. 10, 208. doi:10.5772/56370

Rhudy M, Gu Y, Napolitano MR (2013b) Does the unscented Kalman filter converge faster than the extended Kalman filter? A counter example. AIAA guidance navigation and control conference, Boston, Aug 2013

Rhudy M, Gu Y, Gross J, Gururajan S, Napolitano MR (2013c) Sensitivity analysis of extended and unscented Kalman filters for attitude estimation. AIAA J Aerosp Inf Syst 10(3):131–143

Rugh J, Shamma J (2000) A survey of research on gain scheduling. Automatica 36:1401–1425

Rushby J (1995) Formal methods and their role in digital systems validation for airborne systems. NASA Contractor report 4673, Aug 1995

Rushby J (2009) Software verification and system assurance. In: 7th IEEE international conference on software engineering and formal methods (SEFM), Hanoi, Nov 2009

Simulink®, The Mathworks Inc., Natick, MA

Slotine J-JE, Li W (1991) Applied nonlinear control. Pearson education. Prentice Hall, Upper Saddle River, New Jersey

Summers E, Chakraborty A, Tan W, Topcu U, Seiler P, Balas GJ, Packard AK (2013) Quantitative local L_2-gain and reachability analysis for nonlinear systems. Int J Robust Nonlinear Control, 23:1115–1135

Tallant GS, Hull RA, Bose P, Johnson T, Buffington JM, Krogh B, Crum VW, Prasanth R (2004) Validation & verification of intelligent and adaptive control systems. IEEEAC paper #1487, Dec 2004

Varga A, Looye G, Moormann D, Grubel G (1998) Automated generation of LFT-based parametric uncertainty descriptions from generic aircraft models. Math Comput Model Dyn Syst 4:249–274

Varga A, Hansson A, Puyou G (2012) Optimization based clearance of flight control laws. Springer, Berlin

Wu F, Packard AK, Becker G (1996) Induced L2-norm control for LPV systems with bounded parameter variation rates. Int J Control 6(9/10):983–998

Xu X, Ulrey M, Brown JA, Mast J, Lapis MB (2013) Safety sufficiency for NextGen: assessment of selected existing safety methods, tools, processes, and regulations. NASA/CR-2013-217801, Feb 2013

Zhou K, Doyle JC (1997) Essentials of robust control. Prentice Hall, Englewood Cliffs, New Jersey

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice Hall, Englewood Cliffs, New Jersey

Vehicle Dynamics Control

Eric Tseng
Ford Motor Company, Dearborn, MI, USA

Abstract

Current prevailing control technology enables vehicle dynamic control through powertrain torque manipulation and individual wheel braking. Longitudinal control can maintain vehicle acceleration/braking capability within the physical limits that the road condition can support, while vehicle lateral control can preserve vehicle steering/handling capability up to the

maximum capacity offered by the road/tire interaction. Since most of these controllers are driver-assist systems, their objective is to retain the vehicle dynamic state in operating regions familiar to drivers. In general, this implies that the controller will keep the tire in its linear region and avoid excessive slipping, skidding, or sliding.

Keywords

Active yaw control; Electronic stability control; Evasive maneuvers; Lateral dynamics; Traction assist; Traction control; Vehicle stability assist

Introduction

Vehicle dynamics control generally refers to the active modification of longitudinal and lateral tire forces and the corresponding dynamics of ground vehicles using sensors and actuators. While it may also include vehicle active or semi-active suspension control (Hrovat 1997), vehicle dynamics control in this entry will focus on traction control – vehicle longitudinal control and electronic stability control – combined vehicle longitudinal and lateral control.

Simply speaking, tire force is generated when there exists a velocity difference between tire tread and the ground, also known as tire slip. As illustrated in Fig. 1, the longitudinal tire force

first grows proportionally with the tire slip, in a so-called linear region, and then saturates as tire slip passes beyond a certain threshold. The figure also shows the coupling effect between longitudinal and lateral tire forces. That is, the available lateral force (as a function of tire slip angle) decreases when the longitudinal tire slip increases, and the available longitudinal force (as a function of tire slip) decreases as the lateral tire slip angle increases. This coupling effect is essential for understanding vehicle dynamics and leads to numerous control applications.

Traction Control

Since vehicle motion relies on the tire/ground interaction, it is important for the purpose of vehicle controllability to maintain tire/road interaction in a linear and predictable way. Anti-lock braking systems (ABS), and traction control (TC) in particular, monitor and control the tire slip so that the longitudinal tire force can best support and balance the corresponding brake torque (during ABS intervention) or driveline torque (during TC intervention) delivered to the wheels. Without the wheel/tire slip control, tire force may saturate, resulting in both the reduction of longitudinal and lateral tire force capacity, with the corresponding reduction in decelerating/accelerating capability, or loss of road grip/lateral tire force capacity.

Vehicle Dynamics Control, Fig. 1 Longitudinal and lateral tire force as a function of tire slip and slip angle
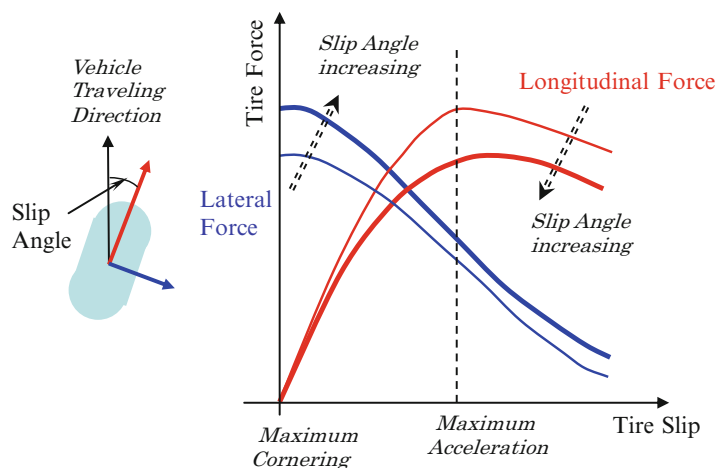
Control Design

The objective of a TC system is to ensure longitudinal tire force capacity while maintaining a good margin on available lateral force road grip (see Fig. 1). Based on the tire force/slip characteristics, this can be achieved by regulating the longitudinal tire slip, roughly defined as the relative velocity between the contact patch of the tire and the road surface. This can be expressed as the difference between the vehicle traveling speed and tire rotational speed, as defined in Eq. 1, according to the Society of Automotive Engineers (SAE), where V is the vehicle speed, ω is the angular speed of the tire, and R is the effective tire rolling radius. The effective rolling radius is defined so that vehicle speed equals the product of R and ω (i.e., $V = R\,\omega$) when there is no torque applied to the wheel and the tire is free rolling.

$$s = \frac{V - R\omega}{V}, \tag{1}$$

At low slip, the longitudinal tire force grows as the slip increases (Carlson and Gerdes 2003), while at high slip it passes its peak and begins to decrease (Deur et al. 2004). High slips occur when wheels are locked during braking or are overspinning during acceleration. A traction control system uses feedback control to regulate wheel/tire slip.

Sensors and Actuators

To regulate driven wheel slips effectively with closed loop control, wheel speed sensors at the non-driven wheels are utilized for vehicle speed estimation (V in Eq. 1). In the case of all-wheel drive or four-wheel drive systems, a longitudinal accelerometer is typically added for the speed estimate. As the amount of desired wheel slip may vary depending on maneuvers, accelerator pedal, steering angle, and yaw rate signals may be used as well. Some systems deploy steering wheel angle and yaw rate sensors for direct signal assessment and signal sharing competency, while others estimate these signals based on the speed difference between left and right wheels, for

subsystem modularity across various vehicle configurations and platforms as well as calibration independency.

Powertrain and brake torque modulation are typically used for actuation to regulate driven wheel slips.

Control System Behavior

Wheel/tire slip targets are typically adjusted based on vehicle driveline configurations as well as vehicle maneuvers. When a vehicle is cornering, a low slip target is generated to assure sufficient margin in lateral tire force capacity. Similarly, rear wheel drive vehicles may warrant a lower slip target than front or all-wheel drive vehicles. When a driver presses hard on the accelerator pedal, the slip target can be raised to accommodate higher acceleration. In addition, the target can be adjusted based on vehicle speed and estimated road available friction, all in an attempt to optimize the longitudinal traction force while keeping sufficient margin on lateral grip (Fodor et al. 1998; Hrovat et al. 2000).

Uniform Friction Surface: (Uniform mu)
Unless equipped with advanced driveline mechanism such as active limited slip differential or torque vectoring differential (Deur 2010), a vehicle is typically equipped with open differential, thus transmitting the powertrain torque evenly to both left and right driven wheels. Since there is no difference between the left and right wheel torque that can be supported by the uniform driving surface, wheel slip regulation can be quite effectively achieved by modulating only the powertrain torque. One successful example of this is Ford's engine-only traction control system, introduced in 2006 on its Fusion and F150 models, which was well received by media experts and customers (Healey 2005).

Nonuniform Friction Surface: (Split mu)
For driving surfaces offering different tire/road characteristics, different wheel torque can be supported on different sides. In this case, a

V

vehicle equipped with an open differential would transmit only the minimum torque (set by the low friction side) to the road. Any additional driveline torque that cannot be supported by the road surfaces results in spinning the wheel on the low friction side. Since open differential transmits equal amount of torque left and right, the additional tire force available on the higher friction side would not be fully utilized with powertrain only actuation. In this case, by applying additional brake torque at the low road friction side, the driveline torque can be balanced at the higher level offered by the high friction side. Care must be taken to avoid aggressive brake application which can cause driveline and/or half shaft oscillations (Fodor et al. 1998; Hrovat et al. 2000).

Control Challenges

Given that road/tire interaction varies with multiple environmental factors, the tire force/slip relationship depicted in Fig. 1 is only a qualitative characterization, and the actual optimal slip for a desired traction force is difficult to accurately establish. While the peak traction force and the corresponding road friction potential (i.e., mu) can be detected once the wheel starts to spin, it is difficult to do so prior to a wheel spin event (Gustafsson 1996).

Since the powertrain actuation is less perceptible yet occasionally sluggish, and the brake application can be fast but intrusive at times, it can be a control challenge to optimize the actuation combination and bandwidth.

If a priori knowledge of friction potential and optimal slip can be learned, detailed powertrain/driveline actuation delay and dynamics can be modeled, and optimal actuation combination and bandwidth can be incorporated; it is conceivable that further improvement in wheel slip and traction control can be achieved, using advanced control approaches such as model predictive control (Borrelli et al. 2006), for example.

Electronic Stability Control

According to the Society of Automotive Engineers (SAE), an electronic stability control system (ESC) is a computer-controlled system that augments vehicle directional stability by applying and adjusting individual wheel braking. It is operational over the full speed range of the vehicle and is capable of monitoring both driver steering input and vehicle yaw rate to limit vehicle understeering and oversteering, as appropriate.

The wide proliferation of ESC in recent years (Van Zanten 2000) across the vehicle fleet has allowed various evaluation studies of its effectiveness in real-world environments. Among them, the United States NHTSA (National Highway Traffic Safety Administration) study (Dang 2004) concluded that ESC reduces fatal single vehicle crashes by 35 %, while single vehicle crashes involving sport utility vehicles (SUVs) are reduced by 67 %. Similar conclusions were arrived in other subsequent studies, including the statement that "Electronic stability control could prevent nearly one-third of all fatal crashes . . . " from the Insurance Institute for Highway Safety organization (IIHS 2006). Many of these effectiveness studies are summarized in a literature review by Ferguson (2007).

Control Design

The objective of an ESC system is to provide vehicle controllability and predictability to assist the driver. This can be achieved by preventing excessive deviations between the intended and actual lateral response of the vehicle, especially during critical maneuvers such as a sudden encounter with a slippery/icy road.

During driving, a driver relies on a mental model of the vehicle's response to his/her steering input developed from previous driving experience. A vehicle model, as described in Eq. 2 and Fig. 2, is often used to describe nominal lateral vehicle behaviors.

**Vehicle Dynamics
Control, Fig. 2** Vehicle
cornering model
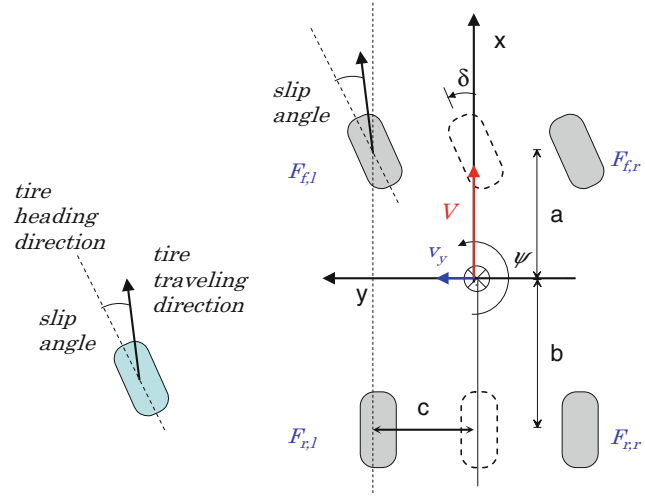
$$m\dot{v}_y = -mV\dot{\psi} + F_{yf}(v_y, V, \delta, F_{xf})$$
$$+ F_{yr}(v_y, V, \delta, F_{xr})$$
$$I\ddot{\psi} = 2aF_{yf} - 2bF_{yr}$$
$$+ c(-F_{xf,l} + F_{xf,r} - F_{xr,l} + F_{xr,r}) \quad (2)$$

where m is the vehicle mass; V and V_y are vehicle longitudinal and lateral velocity, correspondingly; $\dot{\psi}$ is vehicle yaw rate; and δ is the steering angle. Parameters a and b are the distance between vehicle center of gravity to front and rear axle; c is the half track width. F_x and F_y denote the longitudinal and lateral/cornering tire force, with subscript indicating longitudinal (x) or lateral (y) direction, as well as the specific corner of the vehicle (front, rear, left, and right).

Note that the corresponding longitudinal dynamics can be described as

$$m\dot{V} = mv_y\dot{\psi} + F_{xf,l} + F_{xr,l} + F_{xf,r} + F_{xr,r} \quad (3)$$

As the available road friction is not always known, a nominal vehicle lateral response derived from a hi-mu surface may not be feasible and may not best represent a driver's intent. To modify the feedback to best adapt to the road condition, ESC would do one or more of the following: Adjust the driver intended yaw rate according to detected lateral acceleration capability (Tseng et al. 1999), balance between yaw

rate and lateral acceleration error when both are compared to a nominal vehicle model (Manning and Crolla 2007), or balance between yaw rate error and detected excessive sideslip angle (Di Cairano et al. 2013).

Sensors and Actuators

In order to effectively provide vehicle controllability through an embedded controller, ESC systems are equipped with a steering angle sensor, wheel speed sensors, a yaw rate sensor, and a lateral accelerometer. Additional sensors such as a longitudinal accelerometer and a roll rate sensor may be installed to better observe the vehicle dynamic states and provide improved fidelity for estimated vehicle behaviors. The actuators of an ESC system are the individual wheel brakes and powertrain torque.

Control System Behavior

A vehicle can exhibit understeering and/or oversteering behaviors during aggressive lateral maneuvers. Figure 3 illustrates how a vehicle equipped with ESC may provide better controllability.

Understeering – When a vehicle does not turn in as much as desired by the driver (see

Vehicle Dynamics Control, Fig. 3 Vehicle going through a hairpin turn

upper Fig. 3). In this case, the vehicle yaw rate, an ESC measured/monitored signal, would be less than the driver desired value. For example, a vehicle on ice may experience extreme understeering that keeps the vehicle moving straight even when the steering wheel is turned. In this case, ESC applies corrective yaw moment to increase the yaw rate through individual wheel braking. Most of the longitudinal braking force is applied on rear axle inside wheel in the attempt to increase the lateral force capacity on the front axle while decreasing the lateral force capacity on the rear axle. As such, the vehicle experiences not only the yaw moment correction but also the reduction of understeering tendency with ESC brake application.

Oversteering – When a vehicle turns too much, i.e., yaws with a smaller turning radius than the one needed to negotiate the road (see lower Fig. 3). In this case, the vehicle yaw rate would be larger than the driver desires. The vehicle tends to build up a large sideslip angle, resulting in a spinout due to the saturation of rear tire force. In this case, ESC applies corrective yaw moment to decrease the size of yaw rate through individual wheel braking. The longitudinal braking force is applied mostly on the front axle outside wheel to preserve the lateral force capacity on the rear axle and decrease the lateral force capacity on the front axle. As such, the vehicle experiences not only the yaw moment correction but also the reduction of oversteering tendency with ESC brake application.

Evasive Maneuver – During an evasive maneuver, such as an aggressive double lane change, the vehicle may first turn in one direction, followed by an oversteer in the other direction. Due to delay and lag of actuator response in practice, feedforward control is typically used to ensure brake

application would generate corrective yaw moment in the appropriate direction. Further improvement could be possible by using road/traffic preview along with (semi)autonomous intervention based on advanced optimal control such as model predictive control (Falcone et al. 2008), for example.

Skidding and Oversteering – When both front and rear tires experience large tire slip angle, both tire forces are saturated. In this case, the vehicle is operating in a region where the rear tire slip angle can grow rapidly. Unless the front steering is delicately and quickly balanced, the excessive rear tire slip angle could cause the vehicle to spin out. In this case, in addition to applying corrective yaw moment similar to the above oversteering case, ESC may command light braking on all four wheels in an attempt to further slow down the vehicle.

Rollover Mitigation – An ESC system may be extended to further provide a more controllable vehicle behavior and mitigate rollover risks in evasive maneuvers that demand a large and sudden lateral force. For example, Roll Stability Control ^{TM} system introduced at Ford in 2003 monitors vehicle roll behavior in addition to vehicle yaw behavior to assist the driver (Lu et al. 2007).

Control Challenges

In order to best provide the assistance to drivers' desire, it is important to assess the vehicle dynamic state and driver intention with high fidelity. This can be challenging in the presence of various factors that directly influence the vehicle behavior or sensor readings but are not or cannot be directly measured. For example, the road bank angle information is typically unavailable, but it has a direct influence on the lateral accelerometer measurement and could be misinterpreted as a discrepancy between vehicle yaw rate and lateral force (Tseng 2001; Tseng et al. 2007). And despite its criticality in vehicle dynamics control,

the available road surface friction capacity and the vehicle sideslip angle typically cannot be measured (Tseng 2002, Ryu 2002, Ahn et al. 2013). The driver's intent is prescribed by a mental model in the computer, but we cannot directly read the driver's mind. In addition, the controller should detect when a sensor is misbehaving and giving out false or biased readings (Xu and Tseng 2007). While advanced observers have been developed to address these challenges, it is foreseeable that optimization in these areas could further improve the observer fidelity and overall ESC performance.

Cross-References

▶ Lane Keeping
▶ Motorcycle Dynamics and Control

Bibliography

Ahn C, Peng H, Tseng HE (2013) Robust estimation of road frictional coefficient. IEEE Trans Control Syst Technol 21(1):1–13

Borrelli F, Bemporad A, Fodor M, Hrovat D (2006) An MPC/hybrid system approach to traction control. IEEE Trans Control Syst Technol 14(2):541–552

Carlson CR, Gerdes JC (2003) Nonlinear estimation of longitudinal tire slip under several driving conditions. Paper presented in American control conference, Denver, June 2003

Dang JN (2004) Preliminary results analyzing the effectiveness of electronic stability control (ESC) systems, Report no. DOT HS-809-790. National Highway Traffic Safety Administration, Washington, DC

Deur J, Asgari J, Hrovat D (2004) A 3D brush-type dynamic tire friction model, vehicle system dynamics. Int J Veh Mech Mobil 42(3):133–173

Deur J, Ivanoviæ V, Hancock M, Assadian F (2010) Modeling and analysis of active differential dynamics. ASME J Dyn Syst Meas Control 132(6):1–13

Di Cairano S, Tseng HE, Bernardini D, Bemporad A (2013) Vehicle yaw stability control by coordinated active front steering and differential braking in the tire sideslip angles domain. IEEE Trans Control Syst Technol 21(4):1236–1248

Falcone P, Tseng HE, Borrelli F, Asgari J, Hrovat D (2008) MPC-based yaw and lateral stabilization via active front steering and braking. Veh Syst Dyn 46(S1):611–628

V

Ferguson S A (2007) The effectiveness of electronic stability control in reducing real-world crashes: a literature review. Traffic Inj Prev 8(4): 329–338

Fodor M, Yester J, Hrovat D (1998) Active control of vehicle dynamics. Paper presented in 17th AIAA/IEEE/SAE digital avionics systems conference, Seattle

Gustafsson F (1996) Estimation and change detection of tire-road friction using the wheel slip. In: Proceedings of the 1996 IEEE international symposium, computer-aided control system design, Dearborn, pp 99–104

Healey JR (2005) Ford's 2006 Fusion Review, "Traction control on the V-6 test car was just right ...". http://usatoday30.usatoday.com/money/autos/reviews/healey/2005-10-27-fusion_x.htm posted on 27 Oct 2005. Accessed on 30 Aug 2013

Hrovat D (1997) Survey of advanced suspension developments and related optimal control applications. Automatica 33(10):1781–1817

Hrovat D, Asgari J, Fodor M (2000) Automotive mechatronic systems. In: Leondes CT (ed) Mechatronic systems techniques and applications: volume 2 – transportation and vehicular systems. Gordon and Breach Science Publishers, Amsterdam, pp 1–98

Insurance Institute for Highway Safety (IIHS) (2006) Electronic stability control could prevent nearly one-third of all fatal crashes and reduce rollover risk by as much as 80%; effect is found on single- and multiple-vehicle crashes, News Release 13 June 2006. http://www.iihs.org/news/rss/pr061306.html Accessed 30 Aug 2013

Lu J, Messih D, Salib A, Harmison D (2007) An enhancement to an electronic stability control system to include a rollover control function. SAE Trans 116:303–313

Manning WJ, Crolla, DA (2007) A review of yaw rate and sideslip controllers for passenger vehicles. Trans Inst Meas Control 29(1):117–135

Ryu J, Rossetter EJ, Gerdes JC (2002) Vehicle sideslip and roll parameter estimation using GPS. In: Proceedings of AVEC 2002 6th international symposium of advanced vehicle control, Hiroshima

Tseng HE (2001) Dynamic estimation of road bank angle. Veh Syst Dyn 36(4–5):307–328

Tseng HE (2002) A sliding mode lateral velocity observer. In: Proceedings of AVEC 2002 6th international symposium on advanced vehicle control, Hiroshima, pp 387–392

Tseng HE, Ashrafi B, Madau D, Brown AT, Recker D (1999) The development of vehicle stability control at Ford. IEEE/ASME Trans Mechatron 4(2):223–234

Tseng HE, Xu L, Hrovat D (2007) Estimation of land vehicle roll and pitch angles. Veh Syst Dyn 45(5):433–443

Van Zanten AT (2000) Bosch ESP systems: 5 years of experience. SAE Trans 109(7):428–436

Xu L, Tseng HE (2007) Robust model-based fault detection for a roll stability control system. IEEE Trans Control Syst Technol 15(2):519–528

Vehicular Chains

Mihailo R. Jovanović
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

Abstract

Even since the pioneering work of Levine and Athans and Melzer and Kuo, control of vehicular formations has been a topic of active research. In spite of its apparent simplicity, this problem poses significant engineering challenges, and it has often inspired theoretical developments. In this article, we view vehicular formations as a particular instance of dynamical systems over networks and summarize fundamental performance limitations arising from the use of local feedback in formations subject to stochastic disturbances. In topology of regular lattices, it is impossible to have coherent large formations, which behave like rigid lattices, in one and two spatial dimensions; yet this is achievable in 3D. This is a consequence of the fact that, in 1D and 2D, local feedback laws with relative position measurements are ineffective in guarding against disturbances with slow temporal variations and large spatial wavelength.

Keywords

Introduction

Control of vehicular strings has been an active area of research for almost five decades (Levine and Athans 1966; Lin et al. 2012; Melzer and Kuo 1971a, b; Middleton and Braslavsky 2010;

Seiler et al. 2004; Swaroop and Hedrick 1996, 1999; Varaiya 1993). This problem represents a special instance of more general vehicular formation problems which are encountered in the control of unmanned aerial vehicles, satellite formations, and groups of autonomous robots (Bullo et al. 2009; Mesbahi and Egerstedt 2010). Even for the simplest control objective, in which it is desired to maintain a constant cruising velocity and a constant distance between the neighboring vehicles, it has been long recognized that limited information exchange between the vehicles imposes fundamental performance limitations for control design. In particular, look-ahead strategies that rely only on relative spacing information with respect to the preceding vehicle suffer from *string instability*. This phenomenon is characterized by unfavorable amplification of disturbances downstream the vehicular string (Middleton and Braslavsky 2010; Seiler et al. 2004; Swaroop and Hedrick 1996, 1999). In order to avoid this unfavorable spatial application, it is typically required to broadcast the state of the leader to the rest of the formation.

While a precise characterization of fundamental performance limitations in the control of vehicular formations is still an open question, in this article we review recent progress in this area. We begin by highlighting performance limits that arise even in optimally controlled vehicular strings. The LQR problem for vehicular strings was originally formulated in pioneering papers by Levine and Athans (1966) and Melzer and Kuo (1971a, b). These formulations were revisited in Jovanović and Bamieh (2005) where it was shown that the time constant of the optimally controlled closed-loop system increases linearly with the number of vehicles. This reference also employed spatially invariant theory (Bamieh et al. 2002) to demonstrate the lack of exponential stability in the limit of an infinite number of vehicles and to explain the arbitrarily slowing rate of convergence observed in numerical studies of finite strings of increasing sizes. We then summarize a recent result that viewed vehicular strings as the 1D version of vehicular formations on regular lattices in arbitrary spatial dimensions and established fundamental

performance limitations of spatially invariant localized feedback strategies with relative position measurements (Bamieh et al. 2012). It was shown that it is impossible to achieve robustness to stochastic disturbances with only localized feedback in 1D and 2D; yet this can be achieved in 3D. This is a consequence of the fact that, in 1D and 2D, local feedback laws are ineffective in guarding against disturbances with slow temporal variations and large spatial wavelength. An "accordion" type of motion experienced by these spatiotemporal modes compromises formation throughput, and it may occur even in formations that are string stable. Since the phenomenon that we describe also occurs in distributed averaging algorithms, global mean first passage time of random walks, effective resistance in electrical networks, and statistical mechanics of harmonic solids, it is relevant for a broad class of networked dynamical systems.

Optimal Control of Vehicular Strings

We next summarize a linear quadratic regulator problem for vehicular strings (Levine and Athans 1966; Melzer and Kuo 1971a, b) and demonstrate that strategies that penalize only relative position errors between neighboring vehicles yield nonuniform rates of convergence towards the desired formation (Jovanović and Bamieh 2005). In particular, the time constant of the optimally controlled closed-loop system increases linearly with the number of vehicles, and the formation loses exponential stability in the limit of infinite vehicular strings.
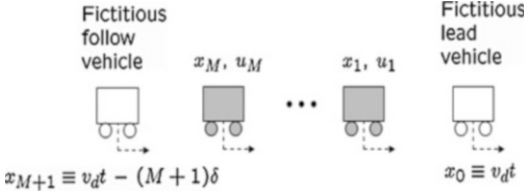
Optimal Control of Finite Strings

A string consisting of M identical unit mass vehicles is shown in Fig. 1a. Each vehicle is modeled as a point mass that obeys the double-integrator dynamics:

$$\ddot{x}_n = u_n, \quad n \in \{1, \dots, M\} \tag{1}$$
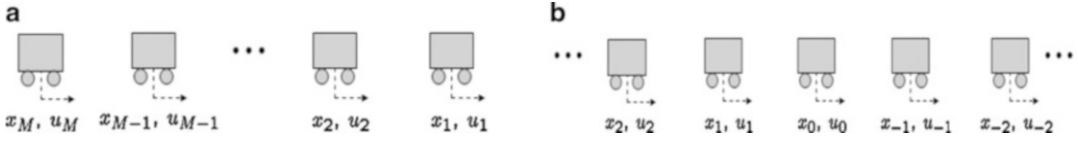
Vehicular Chains, Fig. 1 Finite and infinite strings of vehicles

Vehicular Chains, Fig. 2 Finite string with fictitious lead and follow vehicles

where x_n is the position of the nth vehicle and u_n is the control applied on the nth vehicle. A control objective is to provide the desired constant cruising velocity $\bar{v}$ and to keep the constant distance δ between the neighboring vehicles. By introducing the absolute position and velocity error variables

$$p_n(t) := x_n(t) - \bar{v}t + n\delta$$
$$v_n(t) := \dot{x}_n(t) - \bar{v}, \qquad n \in \{1, \dots, M\}$$

system (1) can be brought into the state-space form (Melzer and Kuo 1971a, b):

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u =: A\psi + Bu \tag{2}$$

where $p := [p_1 \cdots p_M]^T$, $v := [v_1 \cdots v_M]^T$, and $u := [u_1 \cdots u_M]^T$.

Following Melzer and Kuo (1971a, b), fictitious lead and follow vehicles, respectively, indexed by 0 and $M + 1$, are added to the formation; see Fig. 2. These two vehicles are constrained to move at the desired velocity $\bar{v}$, and the relative distance between them is assumed to be equal to $(M + 1)\delta$ for all times. A quadratic performance index that penalizes control effort, relative position, and absolute velocity error variables is associated with system (2):

$$J = \frac{1}{2} \int_0^\infty \left(\sum_{n=1}^{M+1} q_p (p_n(t) - p_{n-1}(t))^2 \right.$$
$$\left. + \sum_{n=1}^M (q_v v_n^2(t) + r\, u_n^2(t)) \right) dt$$
$$= \frac{1}{2} \int_0^\infty (\psi^*(t) Q \psi(t) + u^*(t) R u(t)) dt. \tag{3}$$

The control problem (2) and (3) is in the standard LQR form with the state and control weights:

$$Q := \begin{bmatrix} Q_p & 0 \\ 0 & q_v I \end{bmatrix}, \quad Q_p := q_p T_M, \quad R := rI.$$

Here, T_M is an $M \times M$ symmetric Toeplitz matrix with the first row given by $[\, 2 -1\, 0 \cdots 0\,] \in \mathrm{R}^M$.

We next briefly summarize the explicit solution to the LQR problem (2) and (3) and refer the reader to Jovanović and Bamieh (2005) for additional details. By performing a spectral decomposition of the Toeplitz matrix T_M,

$$T_M = U\Lambda_T U^*, \quad UU^* = U^*U = I$$
$$\Lambda_T = \mathrm{diag}\{\lambda_1(T_M), \dots, \lambda_M(T_M)\}$$
$$\lambda_n(T_M) = 2\left(1 - \cos \frac{n\pi}{M+1}\right), \quad n \in \{1, \dots, M\} \tag{4}$$

the solution to the LQR algebraic Riccati equation can be represented as

$$P := \begin{bmatrix} P_1 & P_0^* \\ P_0 & P_2 \end{bmatrix}, P_0 = U\Lambda_0 U^*, P_2 = U\Lambda_2 U^*,$$
$$P_1 = U\Lambda_1 U^*. \tag{5}$$

Here,

$$\Lambda_0 = \sqrt{rq_p}\, \Lambda_T^{1/2}$$
$$\Lambda_2 = \sqrt{r} \left(2\sqrt{rq_p}\, \Lambda_T^{1/2} + q_v I\right)^{1/2}$$
$$\Lambda_1 = \sqrt{q_p}\, \Lambda_T^{1/2} \left(2\sqrt{rq_p}\, \Lambda_T^{1/2} + q_v I\right)^{1/2}$$

and the eigenvalues of the closed-loop A-matrix are determined by the solutions to the following system of the uncoupled quadratic equations:

$$
\begin{aligned}
s_n^2 + b_n s_n + c_n &= 0, \quad n \in \{1, \ldots, M\} \\
c_n &:= \left(\lambda_n(T_M) q_p / r\right)^{1/2} \\
b_n &:= \left(2c_n + q_v / r\right)^{1/2}.
\end{aligned}
\tag{6}
$$

From the above expression, it can be shown that in large-scale formations, the least-stable eigenvalue of the closed-loop system approaches the imaginary axis at the rate that is inversely proportional to the number of vehicles. As can be seen from the PBH detectability test, this is because the pair (Q, A) gets closer to losing its detectability as the number of vehicles increases. This clearly indicates that the resulting optimal control strategy leads to closed-loop systems with arbitrarily slow decay rates as the number of vehicles increases. As summarized in section "Optimal Control of Infinite Strings," the absence of a uniform rate of convergence for a finite number of vehicles manifests itself as the absence of exponential stability in the limit of infinite vehicular strings.

Optimal Control of Infinite Strings

The LQR problem for a system of identical unit mass vehicles in an infinite string (see Fig. 1b) was originally studied in Melzer and Kuo (1971a). As summarized below, using the theory for spatially invariant linear systems (Bamieh et al. 2002), it was shown in Jovanović and Bamieh (2005) that the resulting LQR controller does not provide exponential stability of the closed-loop system due to the lack of detectability of the pair (Q, A).

The infinite dimensional equivalent of (2) is given by

$$
\begin{bmatrix} \dot{p}_n \\ \dot{v}_n \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_n \\ v_n \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix}
$$

$$
u_n =: A_n \psi_n + B_n u_n, \quad n \in \mathbb{Z} \tag{7}
$$

$$
J = \frac{1}{2} \int_0^\infty \sum_{n \in \mathbb{Z}} \left(q_p (p_n(t) - p_{n-1}(t))^2 \right.
$$

$$
\left. + q_v v_n^2(t) + r\, u_n^2(t) \right) dt \tag{8}
$$
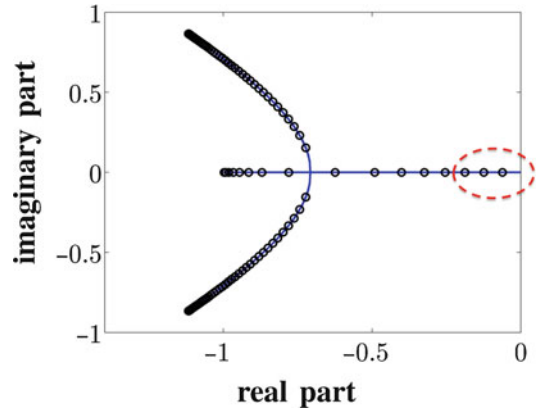
Vehicular Chains, Fig. 3 The spectra of the closed-loop generators in LQR-controlled finite (*symbols*) and infinite (*solid line*) strings of vehicles with $M = 50$ and $q_p = q_v = r = 1$. The closed-loop eigenvalues of the finite string are points in the spectrum of the closed-loop infinite string. As the number of vehicles increases, the number of eigenvalues that accumulate in the vicinity of the stability boundary gets larger and larger

with q_p, q_v, and r being positive design parameters. Spatial invariance over a discrete spatial lattice Z can be used to establish that the solution to the LQR problem does not provide an exponentially stabilizing feedback for system (7). In particular, the spectrum of the closed-loop generator in an LQR-controlled spatially invariant string of vehicles (7) with performance index (8) is given by the solutions to the following θ-parameterized quadratic equation:

$$
\begin{aligned}
s_\theta^2 + b_\theta s_\theta + c_\theta &= 0, \\
c_\theta &:= \left(2(q_p / r)(1 - \cos \theta)\right)^{1/2} \\
b_\theta &:= \left(2c_\theta + q_v / r\right)^{1/2}
\end{aligned}
\tag{9}
$$

where $\theta \in [0, 2\pi)$ denotes the spatial wave number. By comparing (4), (6) and (9), we see that the closed-loop eigenvalues of the finite string are points in the spectrum of the closed-loop infinite string. Furthermore, from these equations it follows that as the size of the finite string increases, this set of points becomes dense in the spectrum of the infinite string closed-loop A-operator. The spectrum of the closed-loop generator, shown in Fig. 3 for $q_p = q_v = r = 1$, illustrates the absence of exponential stability.

Coherence in Large-Scale Formations

Fundamental performance limitations arising from the use of local feedback in networks subject to stochastic disturbances were recently examined in Bamieh et al. (2012). For consensus and vehicular formation control problems in topology of regular lattices, it was shown that it is impossible to guarantee robustness to stochastic exogenous disturbances in one and two spatial dimensions. Yet it was proved that this is achievable in 3D. This phenomenon is a consequence of the fact that, in 1D and 2D, local feedback laws are ineffective in guarding against disturbances with large spatial wavelength, and it has also been observed in global mean first passage time of random walks, effective resistance in electrical networks, and statistical mechanics of harmonic solids. We next briefly summarize the implications of these results for the control of vehicular formations and refer the reader to Bamieh et al. (2012) for details.

Stochastically Forced Vehicular Formations with Local Feedback

Let us consider $M := N^d$ identical vehicles arranged in a d-dimensional torus, Z_N^d, with the double integrator dynamics:

$$\ddot{x}_n = u_n + w_n \qquad (10)$$

where $n := (n_1, \ldots, n_d)$ is a multi-index with each $n_i \in Z_N := \{0, \ldots, N-1\}$, u is the control input, and w is a mutually uncorrelated white stochastic forcing. Each position vector x_n is a d-dimensional vector with components $x_n := \begin{bmatrix} x_n^1 \cdots x_n^d \end{bmatrix}^T$. The control objective is to have the nth vehicle follow the absolute desired trajectory $\bar{x}_n$:

$$\bar{x}_n := \bar{v}t + n\delta \Leftrightarrow \begin{bmatrix} \bar{x}_n^1 \\ \vdots \\ \bar{x}_n^d \end{bmatrix} := \begin{bmatrix} \bar{v}^1 \\ \vdots \\ \bar{v}^d \end{bmatrix} t + \begin{bmatrix} n_1 \\ \vdots \\ n_d \end{bmatrix} \delta.$$

In other words, it is desired that all vehicles move with constant heading velocity $\bar{v}$ while

maintaining their respective position in a Z_N^d grid with spacing of δ in each dimension.

By introducing the position and velocity deviations from the desired trajectory,

$$p_n := x_n - \bar{x}_n, \quad v_n := \dot{x}_n - \bar{v}$$

and by confining our attention to static-feedback policies,

$$u(t) = -\begin{bmatrix} K_p & K_v \end{bmatrix} \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} \qquad (11)$$

equations of motion for the controlled system (10) can be brought into the state-space form

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix}$$
$$w \quad =: A\psi + Bw \qquad (12)$$
$$z \quad = C\psi.$$

Here, p and v are the position and velocity vectors of all vehicles, z is the performance output, and w is the forcing vector.

An Example

In one-dimensional formations with nearest neighbor relative position and velocity measurements, the control acting on the nth vehicle is given by

$$\begin{aligned} u_n(t) = &-k_p^-(p_n(t) - p_{n-1}(t)) \\ &-k_p^+(p_n(t) - p_{n+1}(t)) \\ &-k_v^-(v_n(t) - v_{n-1}(t)) \\ &-k_v^+(v_n(t) - v_{n+1}(t)) \end{aligned} \qquad (13)$$

where $k_p^\pm$ and $k_v^\pm$ are positive design parameters. For a system that evolves over a 1D lattice, the feedback gain matrices K_p and K_v are tridiagonal Toeplitz matrices implying that the closed-loop systems have been effectively converted into a mass-spring-damper system shown in Fig. 4. Figure 5 shows the results of a stochastic simulation for the closed-loop system (12) and (13) with 100 vehicles with desired inter-vehicular spacing $\delta = 20$ and $k_p^\pm = k_v^\pm = 1$. These plots indicate the lack of formation coherence. This is only discernible when one "zooms out" to view
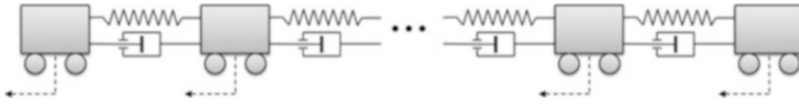
Vehicular Chains, Fig. 4 Finite string of vehicles with a nearest neighbor relative position and velocity feedback
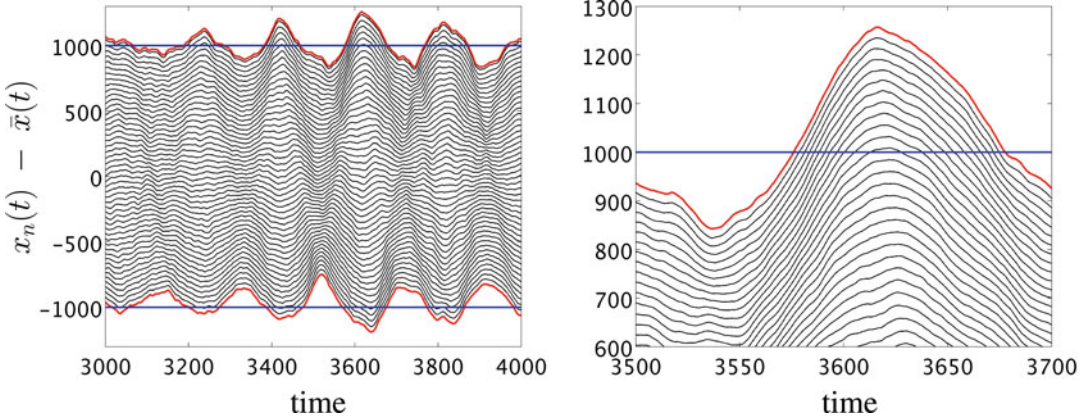
Vehicular Chains, Fig. 5 Position trajectories of a stochastically forced formation with 100 vehicles controlled with nearest neighbor strategy (13). *Left plot* demonstrates accordion-like motion of the entire formation; *right plot* shows that vehicle-to-vehicle distances are relatively well regulated

the entire formation. The length of the formation fluctuates stochastically, but with a distinct slow temporal and long spatial wavelength signature. In contrast, the zoomed-in view in Fig. 5 shows a relatively well-regulated vehicle-to-vehicle spacing. In general, small-scale (both temporally and spatially) disturbances are well regulated, while large-scale disturbances are not. This indicates that a local feedback strategy (13) cannot regulate against large-scale disturbances.

(A3) Reflection symmetry. The interactions between vehicles exhibit mirror symmetry.
(A4) Coordinate decoupling. For $d \geq 2$, control in each coordinate direction depends only on measurements of position and velocity error vector components in that coordinate.

While assumptions **(A3)** and **(A4)** were made to simplify calculations, assumptions **(A1)** and **(A2)** were essential for the developments in Bamieh et al. (2012).

Structural Assumptions
We now list the assumptions on the operators K_p, K_v, and C in (12) under which asymptotic scaling trends summarized in section "Scaling of Variance per Vehicle with System Size" are obtained.
(A1) Spatial invariance. Operators K_p, K_v, and C in (12) are spatially invariant with respect to Z_N^d.
(A2) Spatial localization. The feedback (11) uses only local information from a neighborhood of width $2q$, where q is independent of N.

Performance Measures
We next examine the dependence of the steady-state variance of stochastically forced system (12) on the number of vehicles. In the presence of relative position or velocity measurements, the matrix A in (12) is not necessarily Hurwitz, and the state ψ may not have finite steady-state variance. However, for connected networks, the performance output z that does not penalize the motion of the mean will have finite steady-state variance; this is because the modes of A at the origin will be unobservable from z. The steady-state variance of z,

$$V := \sum_{n \in Z_N^d} \lim_{t \to \infty} \mathcal{E}\left(z_n^T(t) z_n(t)\right) \qquad (14)$$

is quantified by the square of the H_2 norm of the system (12) from w to z, and it can be determined from the solution of the algebraic Lyapunov equation.

We next summarize two different performance measures for stochastically forced vehicular formations.

(P1) Local error. This is a measure of the difference of neighboring vehicles positions from the desired spacing. In 1D, the performance output of the nth vehicle is given by

$$z_n := p_n - p_{n-1}.$$

In d-dimensions, the performance output vector contains as its components the local error in each respective dimension. Since this output involves quantities local to any vehicle within a formation, the corresponding steady-state variance is referred to as a *microscopic performance measure*, V_{micro}.

(P2) Deviation from average. This is a measure of the deviation of each vehicle's position error from the average of the overall position error.

$$z_n := p_n - \frac{1}{M} \sum_{j \in Z_N^d} p_j. \qquad (15)$$

Since this output determines deviation from average, and thereby quantities that are far apart in

the network, the corresponding steady-state variance is referred to as a *macroscopic performance measure*, V_{macro}.

Scaling of Variance per Vehicle with System Size

We next summarize asymptotic bounds for both microscopic and macroscopic performance measures derived in Bamieh et al. (2012). The upper bounds result from simple feedback laws similar to the one given in (13). In the situations where either absolute position or velocity measurement are available, additional terms proportional to p_n and v_n will appear in (13). The lower bounds have been obtained for any linear static feedback control policy satisfying the structural assumptions **(A1)–(A4)** and the following constraint on control variance at each vehicle:

$$\mathcal{E}\left(u_n^T u_n\right) \leq U_{\text{max}}. \qquad (16)$$

Under this constraint, the equivalence between scaling trends of lower and upper bounds can be established. As illustrated in Table 1, the dependence of the asymptotic bounds on the number of vehicles is strongly influenced by the underlying spatial dimension d.

Since the macroscopic performance measure captures how well the formation regulates against large-scale disturbances, the scaling results presented in Table 1 demonstrate that local feedback with relative position measurements is unable to regulate against these large-scale

Vehicular Chains, Table 1 Asymptotic scalings of microscopic and macroscopic performance measures in terms of the total number of vehicles $M = N^d$, the spatial dimensions d, and the control effort per vehicle U_{max}. Quantities listed are up to a multiplicative factor that is independent of M or U_{max}:

Feedback type	V_{micro}/M	V_{macro}/M
Absolute position Absolute velocity	$\frac{1}{U_{\text{max}}}$	$\frac{1}{U_{\text{max}}}$
Relative position Absolute velocity	$\frac{1}{U_{\text{max}}}$	$\frac{1}{U_{\text{max}}} \begin{cases} M & d=1 \\ \log(M) & d=2 \\ 1 & d \geq 3 \end{cases}$
Relative position Relative velocity	$\frac{1}{U_{\text{max}}^2} \begin{cases} M & d=1 \\ \log(M) & d=2 \\ 1 & d \geq 3 \end{cases}$	$\frac{1}{U_{\text{max}}^2} \begin{cases} M^3 & d=1 \\ M & d=2 \\ M^{1/3} & d=3 \\ \log(M) & d=4 \\ 1 & d \geq 5 \end{cases}$

disturbances in 1D. To the contrary, in higher spatial dimensions, local feedback can regulate against large-scale disturbances and provide formation coherence. As shown in Table 1, the "critical dimension" needed to achieve network coherence depends on the type of feedback strategy: dimension 3 for relative position and absolute velocity feedback and dimension 5 for relative position and velocity feedback.

Summary and Future Directions

For stochastically forced vehicular formations in topology of regular lattices, we have summarized fundamental performance limitations resulting from the use of local feedback. Even for formations that are string stable, local feedback is not capable of guarding against slowly varying disturbances with long spatial wavelength in 1D and 2D. The observed phenomenon also arises in distributed averaging and estimation algorithms, global mean first passage time of random walks, effective resistance in electrical networks, and statistical mechanics of harmonic solids. Since performance measures that we used to quantify robustness to disturbances are easily extensible to networks with arbitrary topology and more complex node dynamics, they can be used to evaluate performance of a broad class of networked dynamical systems in future studies.

Cross-References

▶ Averaging Algorithms and Consensus
▶ Flocking in Networked Systems
▶ Networked Systems
▶ Oscillator Synchronization

Bibliography

Bamieh B, Paganini F, Dahleh MA (2002) Distributed control of spatially invariant systems. IEEE Trans Autom Control 47(7):1091–1107
Bamieh B, Jovanović MR, Mitra P, Patterson S (2012) Coherence in large-scale networks: dimension dependent limitations of local feedback. IEEE Trans Autom Control 57(9): 2235–2249
Bullo F, Cortés J, Martínez S (2009) Distributed control of robotic networks. Princeton University Press, Princeton
Jovanović MR, Bamieh B (2005) On the ill-posedness of certain vehicular platoon control problems. IEEE Trans Autom Control 50(9):1307–1321
Levine WS, Athans M (1966) On the optimal error regulation of a string of moving vehicles. IEEE Trans Autom Control AC-11(3):355–361
Lin F, Fardad M, Jovanović MR (2012) Optimal control of vehicular formations with nearest neighbor interactions. IEEE Trans Autom Control 57(9):2203–2218
Melzer SM, Kuo BC (1971a) Optimal regulation of systems described by a countably infinite number of objects. Automatica 7:359–366
Melzer SM, Kuo BC (1971b) A closed-form solution for the optimal error regulation of a string of moving vehicles. IEEE Trans Autom Control AC-16(1):50–52
Mesbahi M, Egerstedt M (2010) Graph theoretic methods in multiagent networks. Princeton University Press, Princeton
Middleton RH, Braslavsky JH (2010) String instability in classes of linear time invariant formation control with limited communication range. IEEE Trans Autom Control 55(7):1519–1530
Seiler P, Pant A, Hedrick K (2004) Disturbance propagation in vehicle strings. IEEE Trans Autom Control 49(10):1835–1842
Swaroop D, Hedrick JK (1996) String stability of interconnected systems. IEEE Trans Autom Control 41(2):349–357
Swaroop D, Hedrick JK (1999) Constant spacing strategies for platooning in automated highway systems. J Dyn Syst Meas Control 121(3):462–470
Varaiya P (1993) Smart cars on smart roads: problems of control. IEEE Trans Autom Control 38(2):195–207

Vibration Control System Design for Buildings

Hidekazu Nishimura
Graduate School of System Design and Management, Keio University, Yokohama, Japan

V

Abstract

This entry reviews vibration control system design of buildings in terms of energy dissipation and seismic isolation including full active control devices and semi-active or passive

devices. Vibration control of buildings subjected to dynamic loadings such as large earthquakes, strong winds, or heavy traffic is one of the most important factors to take into consideration to secure the users. Since energy dissipation is the key technology in vibration control, many kinds of devices have been developed for structural mitigation. Seismic retrofit of buildings is very important because long-period earthquakes occur at considerable distances from the seismic center. Here, we introduce the application of specific devices to the vibration control system design of real buildings, especially in Japan, where there are many earthquakes.

Keywords

Active control; Base isolation; Energy dissipation; Seismic response control; Seismic retrofit; Semi-active control; Vibration control

Introduction

Vibration control of buildings subjected to dynamic loadings such as large earthquakes, strong winds, or heavy traffic is one of the most important factors to consider for the safety of building occupants. Energy dissipation is the key technology in vibration control, and many kinds of devices have been developed for structural mitigation (Soong and Spencer 2002; Spencer and Nagarajaiah 2003). In Japan, the 2011 earthquake occurred on the Pacific coast of Tohoku, prolonged for an extended period to the Tokyo area 400 km away from the seismic center, and caused fatal damages to the buildings of the surrounding areas.

Therefore, seismic retrofitting of buildings is very important in Japan because long-period earthquakes occur at sites far away from their seismic center as well. In particular, old super-high-rise buildings are concentrated in the central ward of Tokyo, Shinjuku, and they have been built on the basis of the theory of flexible structures. During a long-period earthquake, super-high-rise buildings have very

large displacement (about 0.5 m) because of resonance vibration and may need a few minutes to dissipate the structural vibrations. These buildings need to be retrofitted by adding some energy dissipating devices, such as active mass dampers (AMDs), tuned mass dampers, rotating inertial mass dampers, and passive/semi-active base isolation devices.

This entry reviews a vibration control system design of buildings in terms of energy dissipation and seismic isolation including full active control devices and semi-active or passive devices. We introduce the application of specific devices to the vibration control system design of real buildings.

Active Mass Damper

Active, semi-active, or passive mass damper systems have been installed in a large number of high-rise buildings as shown in Fig. 1 (Soong and Spencer 2002; Spencer and Nagarajaiah 2003). Although active mass dampers have historically
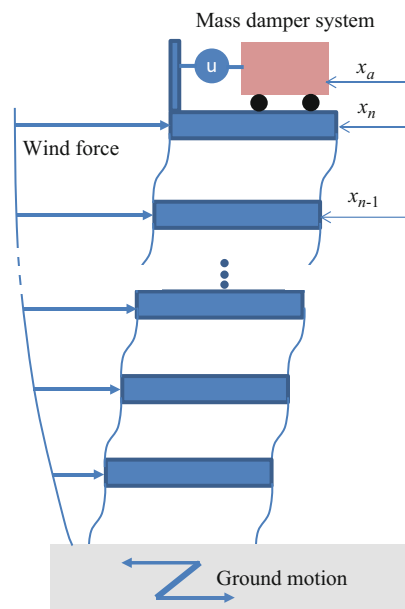
Vibration Control System Design for Buildings, Fig. 1
Active, semi-active, or passive mass damper system installed in an n-storied building subjected to wind force and ground motion

used ball-screw-type actuators, the IHI Corporation has now developed AMDs driven by a linear motor, making the production of a long stroke type easier than that in the case of the ball-screw-type actuator (Koike et al. 2011). Other advantages of using a linear actuator are lesser noise and vibration, lightweight, and compactness. Thanks to these advantages, it is expected that linear motor type AMDs will be installed in existing buildings as seismic retrofitting devices. To avoid reaching the stroke length limit of the actuator because of a large earthquake, a displacement control of the mass is applied. A phase lead compensation in response to the displacement signal is used to preview the mass stroke.

The two AMDs have been installed in the Docomo Tohoku building of Japan in the same direction to improve by 9.5 % the damping ratio of the 1st mode of the translational vibration. The weight of the mass is 20,000 kg, and the total weight of the device is about 25,000 kg. The control experiment was performed by exciting the building with AMDs, and a damping ratio of 11 % was obtained by activating the vibration control with AMDs.

Seismic Retrofitting

Shimizu Corporation modified the super-high-rise building (height = 100 m) in the Shibaura ward of Tokyo, Japan, by installing rotational inertia mass dampers. The rotational inertia mass damper has a mechanism consisting of a ball screw and a rotational inertia mass, with which the relative translational displacement between stories can be changed to rotational motion of the damper to efficiently increase the dissipation of the kinetic energy.

Although in the previous seismic retrofitting many dampers have been distributed in each floor as shown in Fig. 2a, Shimizu Corporation concentrated the rotational inertia mass dampers on the lower floors of the building (e.g., 1–7) as shown in Fig. 2b. The seismic response against the 2011 Tohoku earthquake would now be reduced by about 35 % not only for the maximum displacement but also for the maximum acceleration of the top floor. Moreover, the duration time would become 220 s instead of 400 s. The method of retrofitting super-high-rise buildings is very unique because the lower floors behave as isolation layers of the base isolation system. Although the displacement of the lower floors becomes
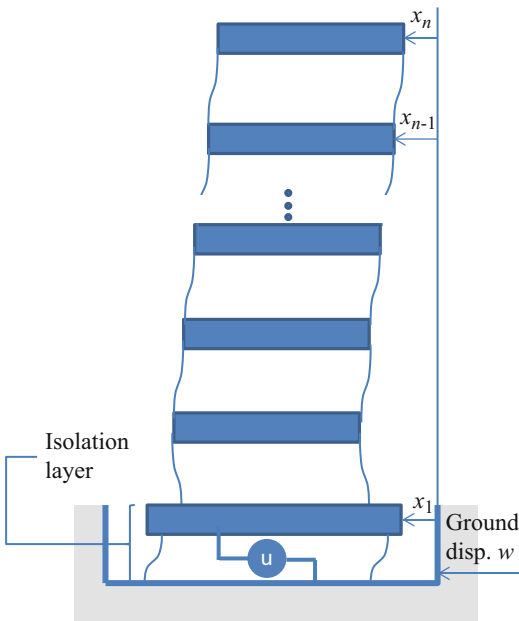

Vibration Control System Design for Buildings, Fig. 2 Seismic retrofitting. (**a**) Distributed energy dissipation device. (**b**) Energy dissipation device concentrated in lower floors

Distributed energy dissipation device

Energy dissipation device concentrated in lower floors
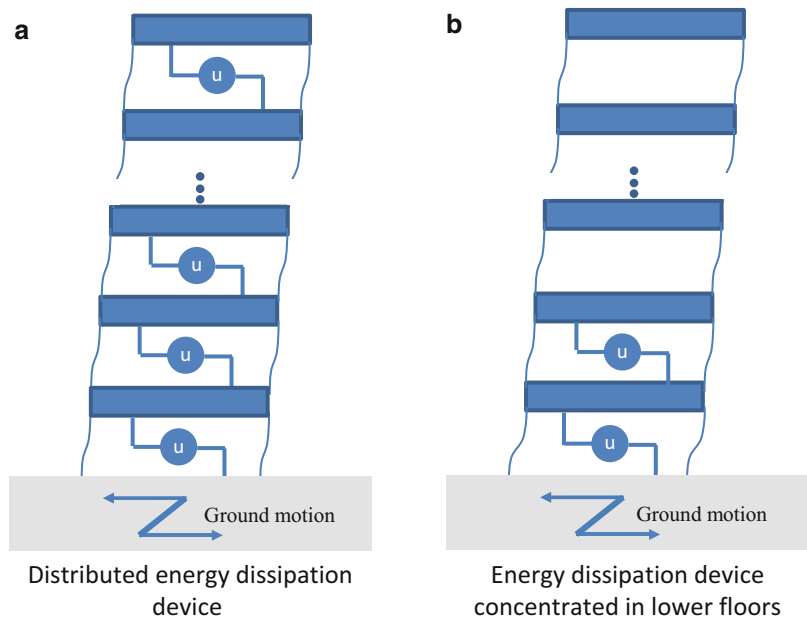
slightly larger than that before the retrofit, the whole building has a good seismic response performance.

Semi-active Base Isolation

The semi-active base isolation system as shown in Fig. 3 has been mounted in a building for the first time in 2000. The building is located on the Yagami campus of the Keio University in Yokohama, Japan, and the base isolation system in two directions consists of eight semi-active hydraulic dampers that can change the damping coefficient in four steps using a controllable orifice. The maximum damping force is 640 kN, while the switching law of damping coefficients is based on the optimal bilinear control theory. The damper is modeled on the lines of the Maxwell model, where a spring and a damper are connected in series, and the objective function on the kinetic energy of the building and the constraint function of the squared damping force are adopted (Yoshida and Fujio 2000).

Vibration Control System Design for Buildings, Fig. 3 Semi-active base isolation system

In 2008, another type of semi-active base isolation system has been installed by Collaboration Complex in the Hiyoshi campus of the Keio University in Yokohama, Japan. The system consists of eight semi-active dampers along with eight conventional hydraulic fluid dampers in each direction of the X–Y axes. While the maximum force of the semi-active damper and the conventional hydraulic fluid damper is about 1,000 kN, the semi-active damper can change the damping coefficient in two steps, high side, 3.68 MNs/m and low side, 1.23 MNs/m. When an earthquake manifests, the high damping coefficient in normal status is switched to the low side. This switch enables the suppression of the acceleration response of the building at the early stages of the earthquake. After the early stage, according to the acceleration response filtered on the isolation layer, the low damping coefficient should be switched to the high side again to avoid the collision of the building with the foundations.

Magneto-rheological (MR) fluid dampers have been studied by many researchers, and in 2001, two 300 kN MR fluid dampers have been installed in Nihon-Kagaku-Miraikan, the Tokyo National Museum of Emerging Science and Innovation. Similarly in 2003, 400 kN MR fluid dampers have been installed in a residential building in Japan (Fujitani et al. 2003).

Although MR fluid dampers have been controlled by various laws (Jansen and Dyke 2000), a gain-scheduled control method was introduced to control the electric current generated by the electromagnet of the MR damper (Nishimura et al. 2002). A system controlled by a damping force is a bilinear control system, where the control input depends not only on the relative velocity but also on the damping coefficient. A virtual semi-active damper model was proposed that is capable of changing the damping coefficient with the valve open ratio, which is assumed to be governed by the input to the dynamics of 2nd-order system. In this device, the optimized variable damping coefficient is determined by the input. Moreover, the valve opening ratio is limited to certain values to constrain the damping force to the maximum value. However, the controllability of the bilinear control system using the variable damping force

depends on the relative velocity of the damper. If the relative velocity equals zero, then the system is uncontrollable. Thus, the systems relative to the positive and the negative sides of the relative velocity are separated. Furthermore, the current–force relationship of the MR damper is considered.

The control method using an MR damper was verified on a 9 m high building-like structure. The structure had four degrees of freedom and a total weight of about 33,000 kg. The MR damper has a maximum force of about 40 kN and its current–force relationship is nonlinear (Watakabe et al. 2008). The experimental results demonstrated a good seismic isolation performance in comparison with the skyhook control. The gain-scheduled control proposed gently varied the damping force according to the input current.

Full Active Base Isolation

Full active base isolation systems have been studied by many researchers (Nishimura and Kojima 1999) who evidenced that they are affected by the saturation of the force generated by the actuator following a large earthquake. The seismic isolation performance should be held even though the force saturation occurred. To control the vibrations, it was proposed to use a hyperbolic function for representing the saturation to smooth the input force (Itagaki and Nishimura 2005).

In 2010, the Obayashi Corporation implemented the active base isolation system in real buildings (Endo et al. 2011). Two hydraulic actuators are connected to the building through a spring in each direction of the X–Y axes to avoid the transmission of the high-frequency vibration from the actuator to the building. The control system is based on the displacement control of the hydraulic actuator and achieves absolute seismic control. The control force is necessary to eliminate the spring and damper forces in the isolation layer, and the skyhook damper force is added to the control force for the stabilization of the whole system.

A trigger mechanism using a friction damper is equipped with serial hydraulic actuators and can avoid the transmission of the excess input force from the actuator to the building. If the excess input force is generated from the actuator in fail, the friction damper can absorb a force of about 1,000 kN so as not to damage the building and the actuator itself. The maximum force of the hydraulic actuator is 1,100 kN, the maximum displacement of the hydraulic actuator is 200 mm, the maximum displacement of the lead–rubber bearing is 500 mm, the maximum displacement of the trigger mechanism with the friction damper is 750 mm, the spring constant of the connected spring is 16,300 kN/mm, and the maximum stroke is 58 mm. Compared to passive isolation, simulations demonstrated that the base isolation system performed well, especially during earthquakes with maximum acceleration less than $200 \, cm/s^2$.

Summary and Future Directions

Seismic retrofitting may become increasingly important for protecting buildings from large and long-period earthquakes. The optimization of the structural mitigation as a whole system must be the objective of future studies aiming to achieve an effective energy dissipation and seismic isolation of buildings. Energy harvesting from vibration control or three-dimensional isolation devices will draw attention in the near future.

Cross-References

▶ H-Infinity Control
▶ Linear Quadratic Optimal Control
▶ LMI Approach to Robust Control
▶ Modeling of Dynamic Systems from First Principles
▶ Stochastic Linear-Quadratic Control

Recommended Reading

Vibration control system design for buildings has been summarized in many journal papers over the last several years. Spencer and Sain

(1997), Soong and Spencer (2002), and Spencer and Nagarajaiah (2003) discuss applications of vibration control systems to buildings or bridges to support infrastructures. Rossetto and Duffour (2012) and Saatcioglu (2012) discuss earthquake-resistant design and structural mitigation of earthquakes with structural control including with passive devices.

Bibliography

Endo F, Yamanaka M, Watnabe T, Kageyama M, Yoshida O et al (2011) Advanced technologies applied at the new "Techno Station" building in Tokyo, Japan. Struct Eng Int 21(4):508–513

Fujitani H, Sodeyama H et al (2003) Development of 400 kN magnetorheological damper for a real base-isolated building. In: Proceedings of the SPIE 5052, smart structures and materials 2003: damping and isolation, San Diego, p 265

Itagaki N, Nishimura H (2005) Disturbance-accommodating gain-scheduled control taking account of actuator saturation. In: Proceedings of the 2005 IEEE conference on control applications, Toronto, 28–31 Aug 2005

Jansen LM, Dyke SJ (2000) Semi-active control strategies for MR dampers: a comparative study. J Eng Mech 126(8):795–803

Koike Y, Imaseki M, Kazama M (2011) Vibration control using rail-guided full-active mass dampers and the application thereof to high-rise buildings. In: Proceedings of the 5th international symposium on wind effects on buildings and urban environment, Tokyo

Nishimura H, Kojima A (1999) Seismic isolation control for a buildinglike structure. IEEE Control Syst Mag 19(6):38–44

Nishimura H et al (2002) Semi-active vibration isolation control for multi-degree-of-freedom structures. In: ASME 2002 pressure vessels and piping conference, seismic engineering, Vancouver, vol 2, Paper no PVP2002-1446, pp 189–196

Rossetto T, Duffour P (2012) Earthquake resistant design. In: Bobrowsky P. (ed.), Encyclopedia of natural hazards, Springer-Verlag Berlin Heidelberg, pp 1–13, 12 Oct 2012

Saatcioglu M (2012) Structural mitigation. In: Bobrowsky P. (ed.), Encyclopedia of natural hazards, Springer-Verlag Berlin Heidelberg, pp 1–25, 17 Sep 2012

Soong TT, Spencer BF Jr (2002) Supplemental energy dissipation: state-of-the-art and state-of-the practice. Eng Struct 24:243–259

Spencer BF Jr, Nagarajaiah S (2003) State of the art of structural control. J Struct Eng 2003: 845–856

Spencer BF Jr, Sain MK (1997) Controlling buildings: a new frontier in feedback. IEEE Control Syst Mag 17(6):19–35

Watakabe M, Inoue N, Nishimura H et al (2008) Response control performance of semi-active isolation system using the GS control for a multi-degree-of-freedom structure with magneto-rheological fluid damper. J Struct Constr Eng 73(628):875–882 (in Japanese)

Yoshida K, Fujio T (2000) Semi-active base isolation for a building structure. Int J Comput Appl Technol 13(1/2):52–58

W

Walking Robots

Ambarish Goswami
Honda Research Institute, Mountain View,
CA, USA

Abstract

This article presents an overview of mobile "walking" robots that use their legs to move from one place to another. Walking robots represent a fascinating class of machines which holds the potential for breakthrough applications and inspires multidisciplinary research with rich scientific content. The key feature that separates walking robots from all other classes of mobile robots is their ability to explore unprepared surfaces using discrete footholds. In this respect, these robots are truly the machine counterparts of biological land animals.

Keywords

Balance; Fall; Gait; Humanoid robots

Introduction

The adventure of modern robotics is generally considered to have started from the middle of the twentieth century (International Federation of Robotics 2011). During the first few decades of this new journey, robots were not mobile. Somewhat similar to trees, these so-called "arm" manipulator robots were securely rooted to the ground. The free end of these robots typically consisted of an end-effector "hand" with which a number of mostly manufacturing-related tasks, such as welding, spray-painting, and pick-and-place operations, were performed. Life was simple, if a bit boring. However, from the end of the 1960s, this started to change.

Fiction writers had earlier imagined a variety of mobile robots such as in "I, Robot" (Asimov 1950), Otho (Hamilton 1940), and Maria (Malone 2004). Scientists and engineers also ventured to build a number of quite sophisticated machines such as the General Electric experimental "walking truck" quadruped robot by Mosher shown in Fig. 1 and the Sparko and Elektro by Westinghouse (http://en.wikipedia.org/wiki/Elektro). However, they were not considered truly autonomous in the sense we describe modern robots. Some of the major personalities who are primarily responsible for forever transforming the state of stationary existence of robots and giving them intelligent mobility are Profs. I. Kato, M. Vukobratovic, and R. McGhee, followed by Prof. M. Raibert.

Because walking robots used legs for locomotion, they immediately became the mechatronic cousins to the entire range of biological legged creatures, starting from tiny creatures to large animals. Indeed, today we have robotic versions
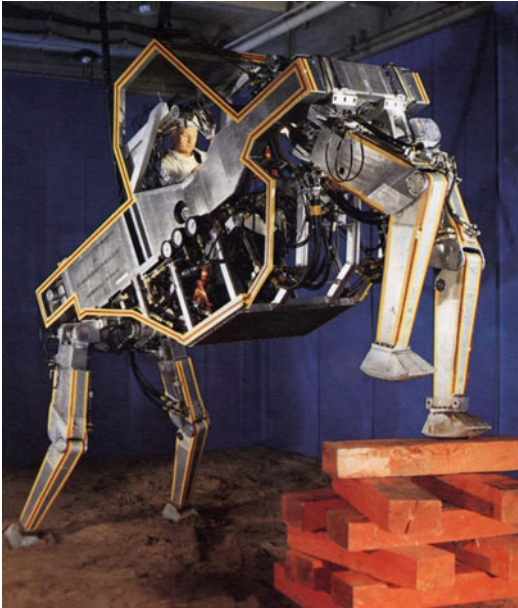
Walking Robots, Fig. 1 GE "walking truck" developed by Mosher

Walking Robots, Fig. 2 Adaptive suspension vehicle (ASV), Ohio State University

of spiders and cockroaches, geckoes and lizards, dogs and cheetah, and even humanoids. We have seen very large robots such as the ASV (Waldron and McGhee 1986) shown in Fig. 2 and the Dante (Bares and Wettergreen 1999), shown in Fig. 4. We have also seen single-legged robots, which even Mother Nature has not considered creating so far.

Early History

The early researchers whom we mentioned above started paving the way for walking robots. These robots walked with their legs, explored their own environments, and sometimes even ventured outside. Once these walking robots started appearing on the scene, life was never the same.

Prof. Kato pioneered walking robot research at Waseda University (Japan) through a series of remarkable biped humanoid robots, of which WL-5 is credited with genuine bipedal walking and WL-6 with displaying the first dynamic gait. At the same time, Prof. Vukobratovic was conducting research activities in exoskeleton and other areas at the Mihailo Pupin Institute (former Yugoslavia). He was instrumental in formalizing the concept of dynamic balance using the zero moment point (ZMP) concept (Sardain and Bessonnet 2004; Vukobratović and Juričić 1969), which is used to this day. In the USA, Prof. McGhee conducted path-breaking research on computer-controlled machines at the Ohio State University. He created the Ohio hexapod and later, with colleague Prof. Ken Waldron, developed the truly spectacular Adaptive Suspension Vehicle (ASV) hexapod.

Prof. Raibert started building robots in the USA, first at Carnegie Mellon University and then at Massachusetts Institute of Technology (Raibert 1989). With his colleagues, he created a series of robots, which, unlike their stationary predecessors, were characteristically full of energy. Situation permitting, they would occasionally deviate from conventional walking and running and would burst into aerial somersaults and other acrobatic motions. Prof. Raibert continues to actively shape the field of walking robots to the present day; his company Boston Dynamics (recently acquired by Google Inc.) has introduced a number of high-performance robots, such as LittleDog, BigDog, RHex, Petman, and Atlas.

The hardware, sensing, and control aspects of walking robots were steadily gaining sophistication during the 1990s. However, except for the new appreciation of walking dynamics

in the study of passive bipedal gait (McGeer 1990), there was no unexpected leap in the world of walking robots. This changed in 1996 when Honda publicly announced the humanoid robot P2, the result of their robotics project, till then unknown to the outside world. This was to be superseded by the P3 robot and then the ASIMO humanoid robot project in 2000, which became another important event in the humanoid robot history.

Characteristics of Walking Robots

Compared to other forms of land locomotion, legged walking possesses the distinct capability of locomotion using discrete footholds (Raibert 1989). Unlike wheeled mobile robots or cars, walking robots do not need a continuous prepared surface such as paved road, trail, or track in order to travel. By virtue of this single feature, a vast extent of land surface, which is not accessible to wheeled robots, opens up to walking robots. Indeed, at least in principle, walking robots are able to reach almost any location, on earth and on other planets, wherever human and other legged creatures can go.

Legged locomotion is natural to terrains where the only means of locomotion must be through the use of unstructured footholds, which can be irregularly spaced both horizontally and vertically. Due to the unique design of the leg, legged creatures can largely isolate the "payload" or the upper body from the geometric details of the terrain profile during locomotion. Both for biological creatures and for walking robots, this brings benefit in the form of significant energy savings. For walking robots this also reduces mechanical stress, vibration, and wear on the system hardware, which makes them suitable for locomotion in rough natural terrain.

In contrast, wheeled robots are typically faster, mechanically less complex, and energetically more efficient. However, these benefits must be supported by very expensive infrastructure overhead. In many places such expenditure is not practical or not even desirable.

Classification of Walking Robots

Walking robots have been built in different sizes and morphologies. These robots have ranged in sizes from small hexapods (Lewinger et al. 2005), medium-sized robots (Fig. 4), and relatively large robots such as the BigDog (Raibert et al. 2008) from Boston Dynamics and Toyota iWalk (Fig. 4) and also a few giant robots such as Dante (Bares and Wettergreen 1999) and Ambler (Fig. 4) from CMU and the ASV (Waldron and McGhee 1986) from OSU. With further miniaturization, it is conceivable that we will see even smaller walking robots in the future with unanticipated and surprising application domains. One can also imagine gigantic walking robots in potential applications in large construction sites such as in bridge, building, or ships, but we have not started seeing them just yet.

In terms of the number of legs, we have already seen monopods, Figs. 3b and 4a; bipeds, Fig. 8a–c; tripod, Fig. 4b; quadruped, Fig. 4a, b; hexapods, Figs. 4c, d and 2; octopod, Fig. 4e; and "centipede" robots with many legs, Fig. 4f.

Other than monopods, robots with odd-numbered legs are curiously absent in this list. Creatures with odd-numbered legs are also not found in nature. It is not clear if an engineering rationale is present behind this trend or the biological inspiration is simply missing for the creators of legged robots.

In addition to size and morphology, walking robots can be classified in terms of the number and types of leg joints, type of gait (e.g., walking or running), or the domain of movement. The next section is devoted to the humanoid robots, which is perhaps the most popular class of walking robots.

Humanoid Robots

Humanoid robots belong to a unique class of two-legged walking robots that has a special place in the popular psyche. These robots are the subject of special affection and fascination due to their similarity with human beings. In fact,

W
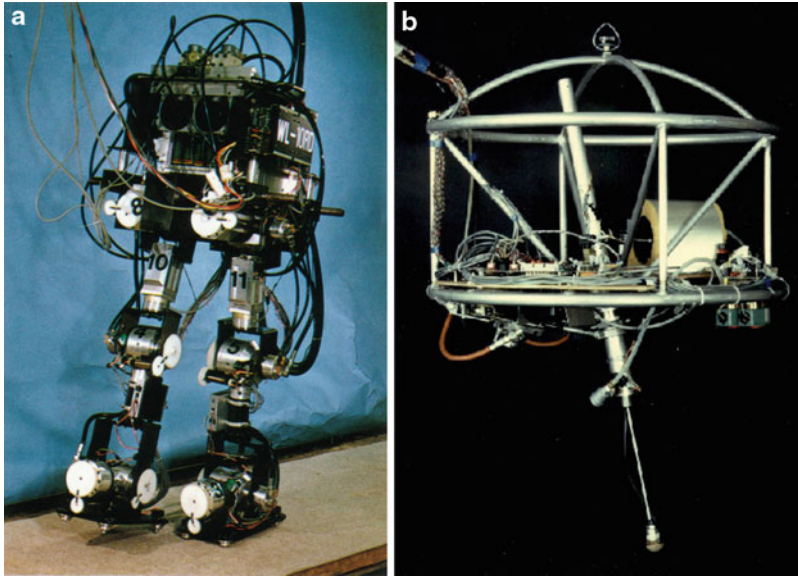
Walking Robots, Fig. 3 Early walking robots: (**a**) Waseda WL-10 (Image courtesy Atsuo Takanishi) and (**b**) one-legged robot (Image courtesy of Boston dynamics)

humanoid robots might be the original inspiration behind the entire field of robotics and perhaps also its ultimate goal. Being perpetually inspired by movies and novels, a long-standing dream of the human has been to create a mechatronic replica of themselves, the human, which will be fully general-purpose endowed with all human functionalities except perhaps the full independence of thought and action.

Humanoid robots exist in different sizes, including smaller robots such as NAO (Gouaillier et al. 2009), HOAP, and QRIO (Ishida et al. 2004) and life-sized robots such as HRP, HUBO, and ASIMO. Despite their differences, these robots bear a close resemblance to the kinematic design and proportions of a human being and share a common human-mimicking morphology. Indeed, the perceived similarity between humanoid robots and the human is so close that we routinely describe aspects of such robots using anthropomorphic terms. Terms like head, arm, hand, leg, thigh, shank, ankle, spine, gait, stumble, fall, facial expression, and even emotion are hardly ever used to describe any other man-made device. Some popular humanoid robots are shown in Fig. 9.

At current technical level, humanoid robots cannot compete in their actual utility with robots such as Roomba the vacuum cleaner, the bomb-sniffing robot, or the huge population of fully active and cost-effective welding and spray-painting robots. Yet, our fascination with humanoids remains as strong as ever, and novel applications of such robots are continuously being explored (Fig. 7). Humanoid robots are currently considered in roles of educators (Falconer 2013; Yamasaki and Nakagawa 2006), dance partners (Kosuge 2010), waiters, babysitters, companions for autistic children or for seniors (Robins et al. 2012), security, or emergency response team. Curiously, the functionality of walking is not relevant or central to many of these roles.

Dynamic Equations of Walking Robots

The dynamic equations of a walking robot can be expressed in the following form:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_g(q) = \Gamma + \Gamma_c + \Gamma_{\text{ext}}, \tag{1}$$
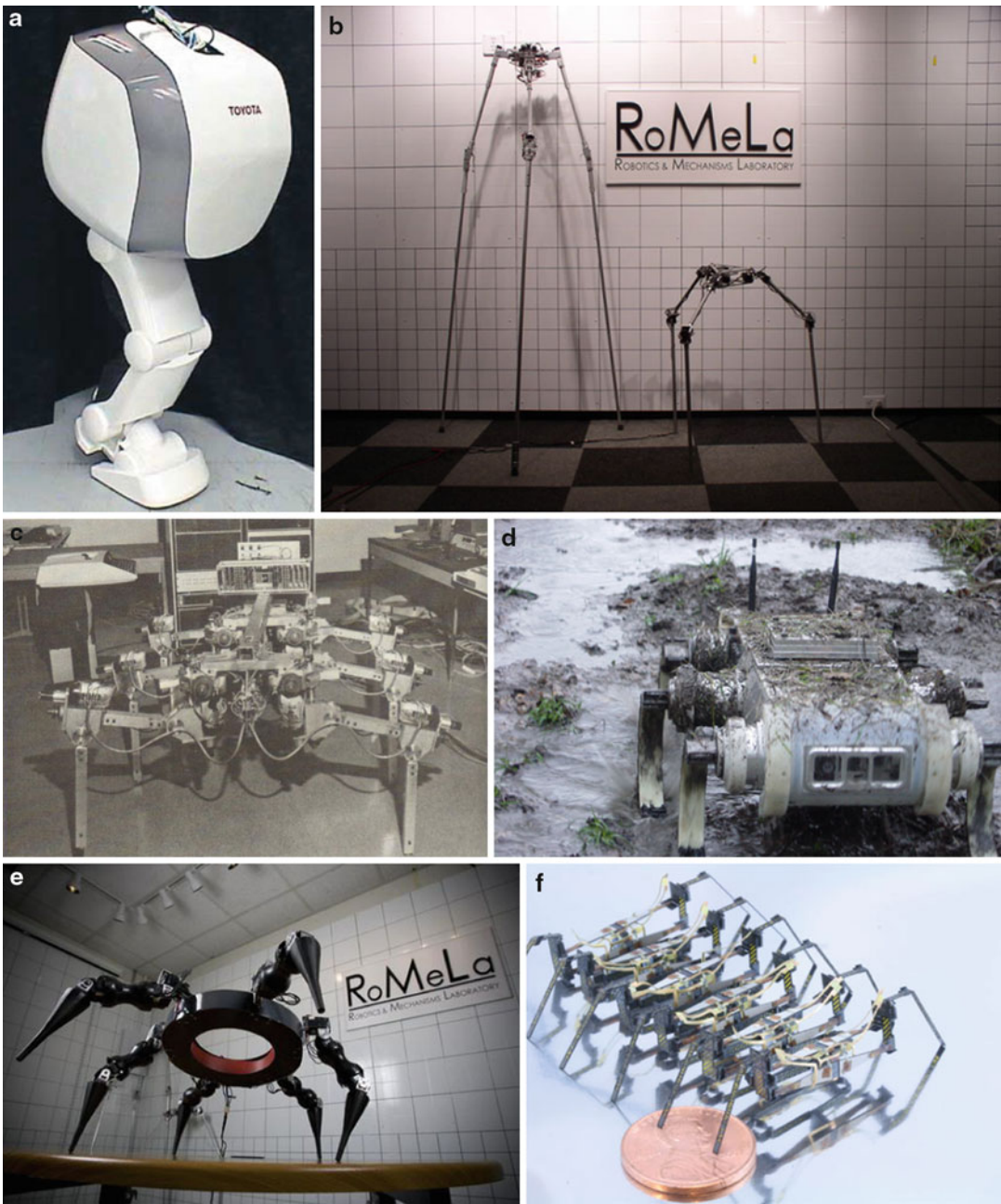
Walking Robots, Fig. 4 Walking robots with different number of legs: (**a**) monopod, Toyota hopping robot; (**b**) tripod, STriDER, RoMeLa (Image courtesy of Dennis Hong); (**c**) large hexapod, McGhee, OSU; (**d**) RHex (RHex robot image courtesy of Boston Dynamics); (**e**) octopod, Spider, RoMeLa (Image courtesy of Dennis Hong); and (**f**) many legs, centipede, Harvard

W
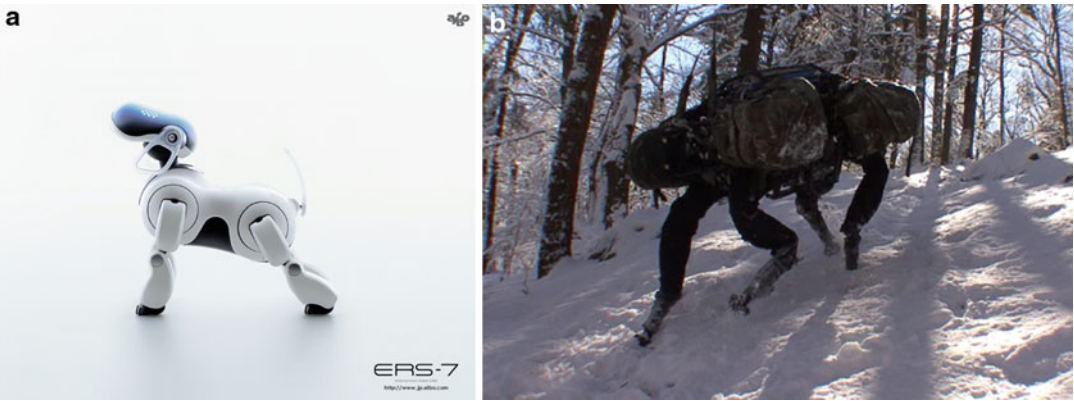
Walking Robots, Fig. 5 Two quadruped robots: (**a**) Sony Aibo (Image courtesy of Sony) and (**b**) BigDog robot (Image courtesy of Boston Dynamics)

Walking Robots, Fig. 6 Large walking robots: (**a**) Dante II, CMU; (**b**) Ambler, CMU; and (**c**) John Deere Walking Tractor
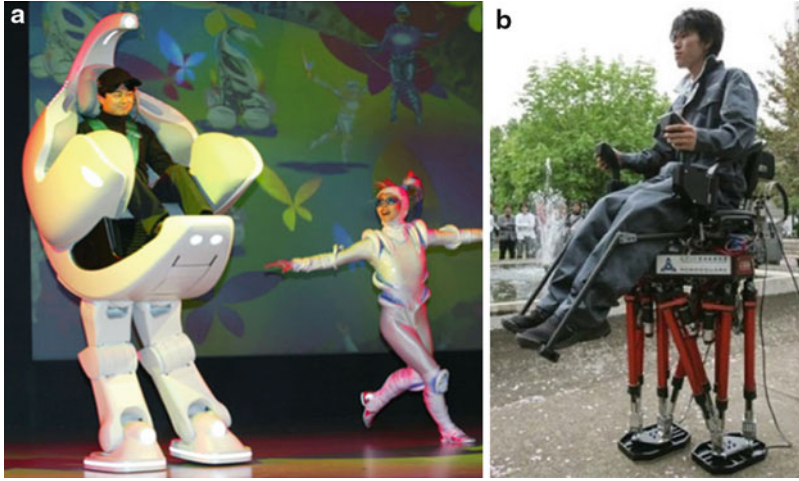
Walking Robots, Fig. 7 Novel application of walking robots: human-carrying "chair" robots, (**a**) iWalk of Toyota and (**b**) WL-16RV multi-purpose biped locomotor from Waseda University (Image courtesy of Atsuo Takanishi)

where q is the vector of the robot's generalized coordinates, which contains the world frame transformation matrix of its base link and all its joint angles. The generalized velocity vector is expressed as $\dot{q} = [v_B \quad \dot{\theta}]^T$ where v_B is the base velocity and $\dot{\theta}$ is the vector of joint velocities. Additionally, H is the joint-space inertia matrix; C is the matrix of Coriolis, centrifugal, and gyroscopic terms; and τ_g is the vector of gravity terms. Finally, $\Gamma = [0 \quad \tau]^T$ is the joint torque vector, $\Gamma_c = J_c^T f_c$ is the joint torque resulting from the contact forces f_c such as from the ground, and $\Gamma_{ext} = J_e^T f_e$ is the joint torque due to external interaction forces f_e.

The contact conditions which the robot must satisfy can be written in the form of Eq. 2. The physical constraints due to ground friction, center of pressure (CoP) condition (explained subsequently), torque limits, etc., can be expressed as in Eq. 3

$$J_c(\ddot{q}) = b(q, \dot{q}), \qquad (2)$$

$$A[\ddot{q} \quad \tau \quad f_c]^T \leq b(q, \dot{q}), \qquad (3)$$

The friction condition ensures that the robot feet do not slide on the ground, and the CoP condition corresponds to maintaining the resultant of the ground reaction force (GRF) within the perimeter of the support polygon (Sardain and Bessonnet 2004) so that toppling is prevented.

Some of the generalized coordinates of the robot, specifically those which describe the base link of the robot to the world frame, are not powered, as apparent from the joint torque vector representation $\Gamma = [0 \quad \tau]^T$, in Eq. 1. In other words, the robot is called *underactuated*. In fact, *all* walking robots are underactuated, and it is one of the central characteristics that sets these robots apart from other robots. Underactuation plays a very important role in the dynamics, motion planning, and control of walking robots (Chevallereau et al. 2005).

Balance and Stability

Even after several decades of research, balance maintenance has remained one of the most important issues of walking robots and especially of humanoid robots. Although the basic dynamics of balance are currently understood (Sardain and Bessonnet 2004; Vukobratović and Juričić 1969), robust and general controllers that can deal with discrete and nonlevel foot support as well as large, unexpected, and unknown external disturbances such as from a moving support, a slip, and a trip have not yet emerged.

Walking Robots, Fig. 8
Two well-known
human-sized humanoid
robots: (**a**) ASIMO,
Honda. (**b**) HRP-2, AIST
(Image courtesy of AIST).
(**c**) HUBO, Korea
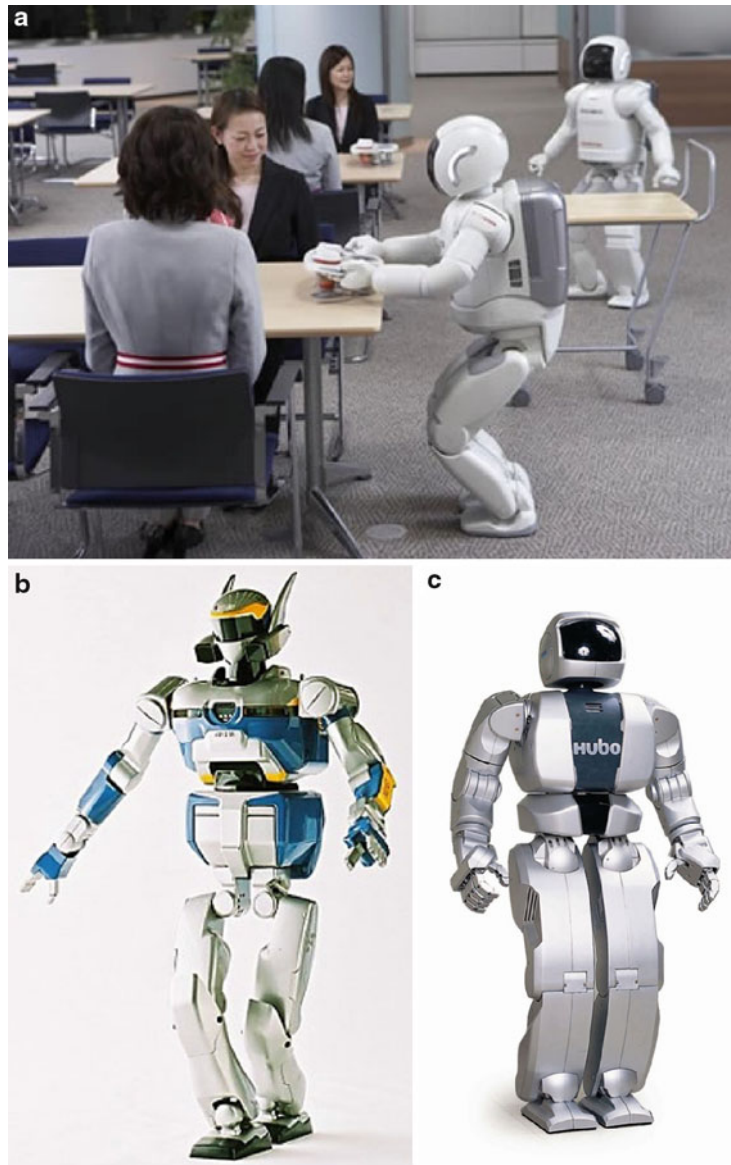
In comparison with the elegance and versatility of human balance, present-day humanoid robots appear quite deficient.

Balance generally refers to the ability of a walking robot to maintain a sustained gait with a reasonably upright posture without falling (Kajita and Espiau 2008). Robot gait can be static or dynamic. A robot with a static gait would continue to stay upright even if its joints were suddenly frozen. Static gait and movement under static balance are safe but are slow and lacks elegance. A dynamic gait is fluid and natural looking as it harnesses and exploits the inertial characteristics of the physical robot. However, the robot must be in motion for it to sustain an upright stature. Suddenly locking the joints may cause a fall.

The location and the nature of the resultant GRF on the support polygon of the robot have been traditionally used to interpret the dynamic state of the robot's movement. The point where the resultant GRF acts on the robot is called its

Walking Robots, Fig. 9 Three popular humanoid robots: (**a**) AIST HRP-4 (Image courtesy of AIST), (**b**) Toyota Partner Robot, and (**c**) Waseda University Wabian (Image courtesy of Atsuo Takanishi)
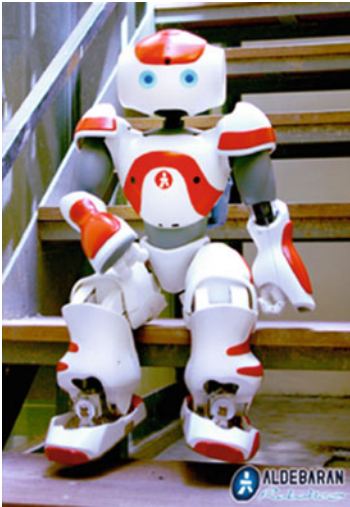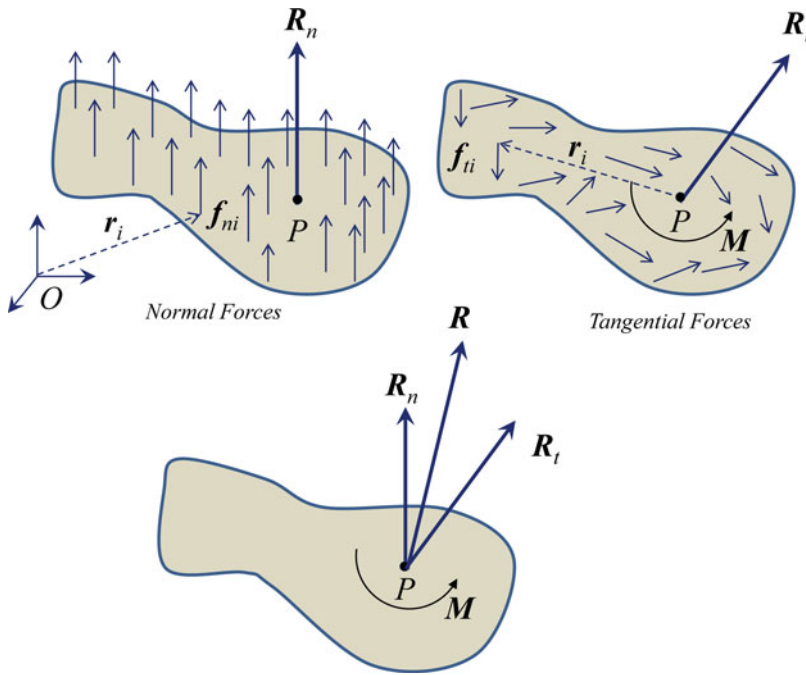

Walking Robots, Fig. 10 Three small humanoid robots: Aldebaran NAO (Image courtesy of Aldebaran), Fujitsu HOAP-2, and Sony QRIO (Image courtesy of Sony)

zero moment point (ZMP), and it is equivalent to the CoP for planar support. Figure 11 explains the concept of CoP.

As shown in Fig. 11, two types of interaction forces act on the foot at the foot/ground interface.

They are the normal forces f_{ni}, always directed upward (Fig. 11, left), and the frictional tangential forces f_{ti} (Fig. 11, middle). CoP, denoted by P, is the point where the resultant $R_n = \sum f_{ni}$ acts. With respect to a coordinate origin

Walking Robots, Fig. 11 Definition of center of pressure (CoP), shown for one foot of a humanoid robot. The idea can be extended to any walking robot, and in a general setting, a single footprint is replaced by the support polygon which is the convex hull of all ground contact of the robot

O, $OP = \dfrac{\sum \boldsymbol{r}_i f_{ni}}{\sum f_{ni}}$, where $\boldsymbol{r}_i$ is the vector to the point of action of force $\boldsymbol{f}_i$ and f_{ni} is the magnitude of $\boldsymbol{f}_{ni}$.

Because of the unilaterality of the foot/ground constraint $\boldsymbol{f}_{ni} \geq 0$, which implies that P must lie within the support polygon. The resultant of the tangential forces may be represented at P by a force $\boldsymbol{R}_t = \sum \boldsymbol{f}_{ti}$ and a moment $\boldsymbol{M} = \sum \boldsymbol{r}_i \times \boldsymbol{f}_{ti}$ where $\boldsymbol{r}_i$ is the vector from P to the point of application of $\sum \boldsymbol{f}_{ti}$. A basic control objective for walking robots is to maintain the CoP within the perimeter of the support polygon.

Safety

Safety is a serious concern that is paramount to any application where robots are likely to coexist in interactive human environments. The power of mobility of walking robots adds to this concern.

Out of a number of possible situations where safety is an issue, one that involves a balance loss and fall is particularly worrisome for walking robots. All walking robots, and in fact all mobile robots, are subjected to this unique "failure" mode. A fall may be caused due to unexpected or excessive external forces, unusual or unknown slipperiness, and slope or profile of the ground, causing the robot to slip, trip, or topple. Fall can also result when the balance controller is partially or fully incapacitated due to an internal failure of the robot involving its sensor or actuator.

Fall can be costly in terms of the damage to the robot and also, depending on the shape and size of the robot, can result in external damage and injury to human.

For humanoid robots, fall is a particularly serious issue (Fujiwara et al. 2002). Humanoid robots, similar to humans, have a larger ratio of CoM height to support area size, which makes them more susceptible to fall, in case of a failure. At the same time, due to their higher CoM, a fall of such robots contains generally higher kinetic energy which is able to cause higher damage and injury.

Summary

Walking robots represent an important class of autonomous machines which can find application in the general area of service robotics. The power of mobility makes these robots uniquely capable of serving in niche need areas such as plant maintenance and security, disaster response, personal companion, and so on. Humanoid walking robots have attracted strong popular fascination, and this has fueled their rapid development. At present it appears that defense-related applications are the most likely to experience practical use of walking robots.

Walking robots possess interesting and complex kinematics and dynamics. Control of such machines, especially with regard to balancing, motion planning, and reactive behavior, is a rich research area that is challenging and demands special skill-sets.

Cross-References

▶ Disaster Response Robot
▶ Redundant Robots
▶ Robot Motion Control
▶ Robot Teleoperation
▶ Underactuated Robots

Recommended Reading

Out of the references listed below, Vukobratović and Juričić (1969) is the earliest paper dealing with bipedal robot balance, and it introduces the concept of ZMP. A very good recent overview of legged robots can be found in Kajita and Espiau (2008). Also of interest is the foundational paper on passive bipedal gait by McGeer (1990).

Bibliography

Asimov I (1950) I, Robot. Bantam Dell, New York, NY

Bares JE, Wettergreen DS (1999) Dante II: technical description, results, and lessons learned. Int J Robot Res 18(7):621–649

Chevallereau C, Westervelt ER, Grizzle JW (2005) Asymptotically stable running for a five-link, four-actuator, planar bipedal robot. Int J Robot Res 24(6):431–464

Falconer J (2013) NAO robot goes to school to help kids with autism. IEEE Specturm, May 2013. http://spectrum.ieee.org/automaton/robotics/humanoids/aldebaran-robotics-nao-robot-autism-solution-for-kids

Fujiwara K, Kanehiro F, Kajita S, Kaneko K, Yokoi K, Hirukawa H (2002) UKEMI: falling motion control to minimize damage to biped humanoid robot. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), St. Louis, pp 2521–2526

Gouaillier D, Hugel V, Blazevic P, Kilner C, Monceaux J, Lafourcade P, Marnier B, Serre J, Maisonnier B (2009) Mechatronic design of NAO humanoid. In: IEEE international conference on robotics and automation (ICRA), Kobe, pp 2124–2129

Hamilton E (1940). Collected Captain Future, Haffner Press, Royal Oak, Michigan

http://en.wikipedia.org/wiki/Elektro

International Federation of Robotics (IFR) (2011) press release. http://www.ifr.org/news/ifr-press-release/50-years-industrial-robots-410/

Ishida T, Kuroki Y, Takahashi T (2004) Analysis of motions of a small biped entertainment robot. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Sendai, pp 142–147

Kajita S, Espiau B (2008) Legged robots. In: Siciliano B, Khatib O (eds) Springer Handbook of Robotics. Springer, Berlin, pp 361–389

Kosuge K (2010) Dance partner robot: an engineering approach to human-robot interaction. In: 5th ACM/IEEE international conference on human-robot interaction (HRI), Osaka

Lewinger WA, Branicky MS, Quinn RD (2005) Insect-inspired, actively compliant hexapod capable of object manipulation. In: Proceedings of the CLAWAR'2005 – 8th international conference on climbing and walking robots, Springer-Verlag Berlin Heidelberg

Malone R (2004) Ultimate robot. DK Publishing, New York

McGeer T (1990) Passive dynamic walking. Int J Robot Res 9(2):62–82

Raibert M (1989) Legged robots. In: Brady M (ed) Robotics science. System development foundation benchmark series. MIT, Cambridge

Raibert M, Blankespoor K, Nelson G, Playter R, The Big-Dog Team (2008) BigDog, the rough-terrain quaduped robot. In: Proceedings of the 17th IFAC world congress, Seoul, pp 10822–10825

Robins B, Dautenhahn K, Dickerson P (2012) Embodiment and cognitive learning – can a humanoid robot help children with autism to learn about tactile social behaviour. Soc Robot Lect Notes Comput Sci 7621:66–75

Sardain P, Bessonnet G (2004) Forces acting on a biped robot. Center of pressure-zero moment point. IEEE Trans Syst Man Cybern 34:630–637

W

Vukobratović M, Juričić D (1969) Contribution to the synthesis of biped gait. IEEE Trans Bio-Med Eng 16(1):1–6

Waldron KJ, McGhee RB (1986) The adaptive suspension vehicle. IEEE Control Syst Mag 6(6): 7–12

Yamasaki F, Nakagawa Y (2006) Education using small humanoid robot. In: Proceedings of the 3rd international symposium on autonomous minirobots for research and edutainment (AMiRE 2005), Fukui, pp 248–253

Wheeled Robots

Giuseppe Oriolo
Sapienza Università di Roma, Roma, Italy

Abstract

The use of mobile robots in service applications is steadily increasing. Most of these systems achieve locomotion using wheels. As a consequence, they are subject to differential constraints that are nonholonomic, i.e., non-integrable. This article reviews the kinematic models of wheeled robots arising from these constraints and discusses their fundamental properties and limitations from a control viewpoint. An overview of the main approaches for trajectory planning and feedback motion control is provided.

Keywords

Differential flatness; Nonholonomic constraints; Nonlinear controllability; Smooth stabilizability

Introduction

Although all robots are, by definition, capable of movement, the expression *mobile robots* is mainly used to indicate robots that can displace their own base by means of some locomotion mechanism. Most often, this consists of a set of wheels. The main advantage of mobile robots over fixed-base manipulators

is their virtually unlimited workspace. As a consequence, such robots are fundamental in service applications, which require increased capabilities of autonomous motion.

More precisely, from a mechanical viewpoint, a *wheeled robot* essentially consists of a rigid body (base) equipped with a system of wheels. This basic arrangement may be complicated, for example, by attaching to the base one or more trailers, or by mounting a manipulator on the base (mobile manipulator).

Any wheeled vehicle is subject to kinematic constraints that in general reduce its local mobility while leaving intact the possibility of reaching arbitrary configurations by appropriate maneuvers. For example, any driver knows by experience that, while it is impossible to move instantaneously a car in the direction orthogonal to its heading, it is still possible to park it anywhere, at least in the absence of obstacles. This peculiar feature makes wheeled mobile robots very challenging from the control viewpoint, and in fact, some recent developments in nonlinear control were triggered by the study of these systems.

Here, we will consider only mobile robots that are equipped with conventional wheels, either orientable or fixed (as the front or rear wheels of a car, respectively). Omnidirectional mobile robots realized using, e.g., Mecanum wheels, are not covered in this article. Indeed, the local mobility of these vehicles is unrestricted, and therefore no special control treatment is necessary.

The most popular wheel arrangement for mobile robots is the *differential drive*, in which two fixed wheels whose axes of rotation coincide are controlled by separate actuators (see Fig. 1). One or more passive (caster) wheels are usually added for statical balance. This wheeled robot is the most agile, in that it can rotate on the spot by applying equal and opposite angular speeds to the wheels. A kinematically equivalent arrangement is the *synchro drive*, in which three orientable wheels are synchronously driven by two motors through mechanical coupling; the first motor provides traction, whereas the second controls the common orientation of the wheels.

Other possible wheel arrangements are those of a tricycle (one steering and two fixed wheels)

Wheeled Robots, Fig. 1 The Pioneer by Adept is a popular differential-drive platform

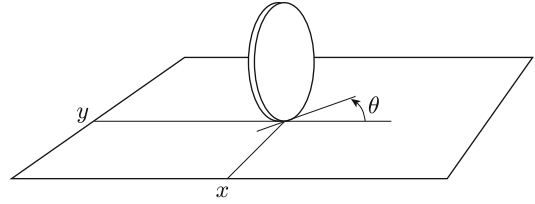**Wheeled Robots, Fig. 2** Generalized coordinates for a single wheel

and of a car (two steering and two fixed wheels). Vehicles of this type are however less common in robotics, due partly to their reduced maneuverability (they have a nonzero turning radius) and partly to their increased mechanical complexity. For example, both these vehicles require a specific device (differential) for distributing traction torque to the driving wheels.

Modeling

The starting point for modeling wheeled mobile robots is the single wheel. This may be represented as an upright disk rolling on the ground. Its configuration is described by three generalized coordinates: the Cartesian coordinates (x, y) of the contact point with the ground, measured in a fixed reference frame, and the orientation θ of the disk plane with respect to the x axis (see Fig. 2). The configuration vector is therefore $q = (x \ \ y \ \ \theta)^T$. The *pure rolling* constraint is expressed as

$$\begin{pmatrix} \sin \theta & -\cos \theta \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = 0 \qquad (1)$$

and entails that, in the absence of slipping, the velocity of the contact point has a zero component in the direction orthogonal to the wheel plane. The angular speed of the wheel around the vertical axis is instead unconstrained.

The kinematic constraint (1) is *nonholonomic*, i.e., it cannot be integrated to a geometric constraint; this may be easily shown using Frobenius theorem, a well-known differential geometry result on integrability of differential forms. An important consequence of this fact is that constraint (1) implies no loss of accessibility in the configuration space of the wheel.

In a single-body vehicle equipped with multiple wheels, the n-dimensional configuration vector q consists of the Cartesian coordinates of a representative point on the robot, the orientation of all independently orientable wheels, plus the orientation of the body if there are fixed wheels. By writing one pure rolling constraint like (1) for each independent wheel, orientable or fixed, and expressing it in the chosen generalized coordinates, one obtains a set of k constraints in the form

$$A^T(q)\dot{q} = 0. \qquad (2)$$

Kinematic constraints of this form (i.e., linear in the generalized velocities) are called *Pfaffian*. In wheeled mobile robots, Pfaffian constraints are in general completely nonholonomic.

The k Pfaffian constraints (2) reduce the number of degrees of freedom (i.e., independent instantaneous motions) of the robot to $m = n - k$. In particular, at each configuration q, the generalized velocities must belong to the m-dimensional null space of matrix $A^T(q)$:

$$\dot{q} = \sum_{j=1}^{m} g_j(q)u_j = G(q)u, \qquad (3)$$

where vectors $g_1(q), \ldots, g_m(q)$ are a basis of $\mathcal{N}(A^T(q))$ and $u = (u_1 \ \ldots \ u_m)^T$ is a coefficient

W

vector. Kinematically admissible trajectories are the solutions of (3), which is called *kinematic model* of the wheeled mobile robot. This model can be seen as a nonlinear dynamic system, with q as state and u as input. In particular, system (3) is driftless and has more state variables than control inputs.

For example, consider the *unicycle*, a rather ideal mobile robot equipped with a single, orientable wheel. The generalized coordinates for this robot are $q = (x\, y\, \theta)^T$, the same as the single wheel, and the vehicle is subject to the rolling constraint (1). One possible kinematic model for the unicycle is then

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega, \qquad (4)$$

where $v = \sqrt{\dot{x}^2 + \dot{y}^2}$ and $\omega = \dot{\theta}$ represent, respectively, the driving and steering velocity of the wheel. Both the differential-drive and the synchro-drive robots are kinematically equivalent to the unicycle, i.e., their kinematic model can be put in the form (3) by properly defining q and u.

Similar to what is done for robot manipulators, the dynamic models of wheeled mobile robots may be derived following the Euler-Lagrange method. The main difference is the presence of the nonholonomic Pfaffian constraints, which give rise to reaction forces expressed via Lagrange multipliers (Neimark and Fufaev 1972).

Structural Properties

The nonholonomic nature of wheeled mobile robots has precise consequences in terms of structural properties of the kinematic model (3).

The first, and most important, is that in spite of the reduced number of degrees of freedom, a wheeled robot is *controllable* in its configuration space; i.e., given two arbitrary configurations, there always exists a kinematically admissible trajectory (with the associated velocity inputs) that transfers the robot from one to the other
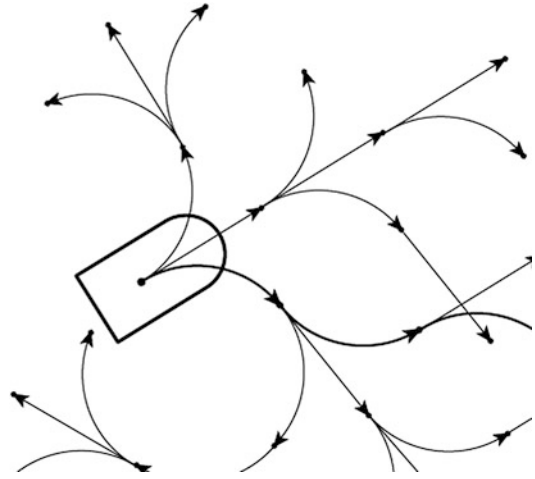
Wheeled Robots, Fig. 3 In spite of its restricted local mobility, a nonholonomic wheeled robot can reach any point in its configuration space

(Fig. 3). Since the kinematic model (3) is driftless, a well-known result (Chow theorem) implies that it is controllable if and only if the accessibility rank condition holds:

$$\dim \bar{\Delta} = n, \qquad (5)$$

where $\bar{\Delta}$ denotes the involutive closure of distribution $\Delta = \{g_1, \ldots, g_m\}$ under the Lie bracket operation. In turn, this is guaranteed to be true in view of the nonholonomy of constraints (2). For example, since the Lie bracket of the two input vector fields in (4) is always linearly independent from them, the kinematic model of the unicycle is controllable.

However, the controllability of wheeled mobile robots is intrinsically nonlinear. In fact, the linear approximation of (3) at any configuration clearly results to be uncontrollable due to the reduced number of inputs. In practice, this means that no linear feedback can stabilize the system at a given configuration. The situation is actually worse: for nonholonomic robots, there exists no continuous time-invariant feedback law that provides point stabilization. This negative result can be established on the basis of a celebrated result on smooth stabilizability of control systems due to Brockett (1983). Note that the result does

not apply to time-varying stabilizing controllers, which may thus be continuous in q.

Another related drawback of wheeled mobile robots is that in general, they do not admit *universal* controllers, i.e., feedback control laws that can asymptotically stabilize arbitrary state trajectories, either persistent or not (Lizárraga 2004). This means that, in principle, tracking and regulation problems in wheeled robots should be addressed using separate approaches.

All the above limitations of nonholonomic systems are established with reference to the kinematic model, but of course, they are passed on to dynamic models. Altogether, they contribute to making the control problem for wheeled mobile robots much more difficult than, for example, for robotic manipulators, which are linearly controllable, smoothly stabilizable and admit universal controllers.

Trajectory Planning

Trajectory planning for wheeled robots is a nontrivial problem, because not all trajectories are feasible – once again, a consequence of nonholonomy. This leads to the necessity of maneuvering, i.e., performing certain specific movements, in order to execute transfer motions.

Most kinematic models of wheeled mobile robots exhibit a property known as *differential flatness* (Fliess et al. 1995): namely, there exists a set of outputs z, called *flat* outputs, such that the state q and the control inputs u can be expressed algebraically as a function of z and its time derivatives up to a certain order σ:

$$q = \varphi\left(z, \dot{z}, \ddot{z}, \ldots, z^{(\sigma)}\right) \qquad (6)$$

$$u = \gamma\left(z, \dot{z}, \ddot{z}, \ldots, z^{(\sigma)}\right). \qquad (7)$$

As a consequence, once an output trajectory $z(t)$ is specified, the associated state trajectory $q(t)$ and control history $u(t)$ are uniquely determined. For example, the unicycle admits $z = (x \ \ y)^T$ as flat outputs. In fact, once a Cartesian trajectory is assigned for the contact point with the ground, the wheel orientation $\theta(t)$ is constrained to be tangent to the trajectory; the associated control input v and ω are then uniquely and algebraically computable from $q(t)$.

Differential flatness is particularly useful for planning. For example, assume that we want to transfer a wheeled mobile robot from an initial configuration q_i to a final configuration q_f. One then computes the corresponding values z_i and z_f of the flat outputs, plus the appropriate boundary conditions, and uses any interpolation scheme (e.g., polynomial interpolation) to plan the trajectory of z. The evolution of the generalized coordinates q, together with the associated control inputs u, can then be computed algebraically from (6–7). The resulting configuration space trajectory will automatically satisfy the nonholonomic constraints (2).

Another approach to nonholonomic trajectory planning is based on the possibility of putting the equations of most wheeled robots into a canonical format known as a 2-input *chained form*

$$
\begin{aligned}
\dot{z}_1 &= w_1 \\
\dot{z}_2 &= w_2 \\
\dot{z}_3 &= z_2 w_1 \qquad\qquad (8) \\
&\ \ \vdots \\
\dot{z}_n &= z_{n-1} w_1
\end{aligned}
$$

by means of a feedback transformation, i.e., a change of coordinates $z = \alpha(q)$ coupled with an input transformation $w = \beta(q)u$. In particular, this is always possible with kinematic models (3) for which $n \leq 4$ and $m = 2$ (e.g., unicycle or car-like robots). Once the system is cast in the form (8), one may use sinusoidal open-loop controls at integrally related frequencies to drive all variables sequentially to their final values (Murray and Sastry 1993). This approach is particularly interesting from a theoretical viewpoint because such control maneuvers achieve motion in the direction of the Lie brackets of the input vector fields.

Note that differential flatness and chained-form transformability are equivalent properties for 2-input nonholonomic mobile robots.

W

Feedback Control

The motion control problem for wheeled mobile robots is generally formulated with reference to the kinematic model (3). For example, in the case of the unicycle (4), this means that the control inputs are directly v and ω, the driving and steering velocities. There are essentially two reasons for taking this simplifying assumption.

First, the kinematic model (3) fully captures the essential nonlinearity of single-body wheeled robots, which stems from their nonholonomic nature. This is another fundamental difference with respect to the case of robotic manipulators, in which the main source of nonlinearity is the inertial coupling among multiple bodies. Second, in mobile robots it is typically not possible to command directly the wheel torques, because there are low-level wheel control loops integrated in the hardware or software architecture. Any such loop accepts as input a reference value for the wheel angular speed, which is then reproduced as accurately as possible by standard regulation actions (e.g., PID controllers). In this situation, the actual inputs available for high-level control are precisely these reference velocities.

Two basic control problems can be considered:
- *Trajectory tracking*: the robot must asymptotically track a desired Cartesian trajectory $(x_d(t), y_d(t))$.
- *Point stabilization*: the robot must asymptotically reach a desired configuration q_d.

From a practical point of view, the most relevant of these problems is certainly the first. This is because mobile robots must be able to operate in unstructured workspaces that invariably contain obstacles. Clearly, forcing the robot to move along (or close to) a trajectory planned in advance reduces considerably the risk of collisions. The point stabilization problem, however, is more difficult and therefore particularly interesting from a scientific perspective. In a certain sense, the relative difficulty of the two problems is reminiscent of human car driving: learning to drive a car along a road is relatively easy, whereas parking poses a greater challenge.

Trajectory Tracking

Several methods are available to drive a wheeled mobile robot in feedback along a desired trajectory. A straightforward possibility is to compute first the linear approximation of the system along the desired trajectory (which, unlike the approximation at a configuration, results to be controllable) and then stabilize it using linear feedback. Only local convergence, however, can be guaranteed with this approach. For the kinematic model of the unicycle, global asymptotic stability may be achieved by suitably morphing the linear control law into a nonlinear one (Canudas de Wit et al. 1993).

In robotics, a popular approach for trajectory tracking is input–output linearization via static feedback. In the case of a unicycle, consider as output the Cartesian coordinates of a point B located ahead of the wheel, at a distance b from the contact point with the ground. The linear mapping between the time derivatives of these coordinates and the velocity control inputs turns out to be invertible provided that b is nonzero; under this assumption, it is therefore possible to perform an input transformation via feedback that converts the unicycle to a parallel of two simple integrators, which can be globally stabilized with a simple proportional controller (plus feedforward). This simple approach works reasonably well. However, if one tries to improve tracking accuracy by reducing b (so as to bring B close to the ground contact point), the control effort quickly increases.

Trajectory tracking with $b = 0$ (i.e., for the actual contact point on the ground) can be achieved using dynamic feedback linearization (Oriolo et al. 2002). In particular, this method provides a one-dimensional dynamic compensator that transforms the unicycle into a parallel of two double integrators, which is then globally stabilized with a proportional-derivative controller (plus feedforward). In contrast to static feedback linearization, no residual zero dynamics is present in the transformed system. However, the dynamic compensator has a singularity when the unicycle driving velocity is zero. This is expected, because otherwise the tracking

controller would represent a universal controller. Note that dynamic feedback linearizability using the x, y outputs is related to them being flat – the two properties are equivalent.

Point Stabilization

The impossibility of stabilizing a nonholonomic mobile robot using continuous pure-state feedback has generated two main directions of research to solve the problem:

- *Discontinuous* feedback, i.e., time-invariant control laws $u = \gamma(q)$, where γ is discontinuous precisely at the configuration that one seeks to stabilize.
- *Time-varying* feedback, in the form $u = \gamma(q, t)$ where γ may or may not be continuous at the desired configuration.

For the unicycle, a well-known stabilizing controller belonging to the first category was designed by Aicardi et al. (1995) by formulating the problem in polar coordinates centered at the goal and then using a Lyapunov-like analysis to establish asymptotic convergence. The controller, once rewritten in original coordinates, turns out to be discontinuous at the goal (not surprisingly). Although this rules out proper stability in the sense of Lyapunov, this controller is effective in that it produces rather natural approach trajectories to the goal.

Continuous time-varying stabilizers in the sense of Lyapunov exist (Samson 1993) but have mainly theoretical interest due to their provably slow (polynomial) rate of convergence; this is a direct consequence of the fact that the linear approximation of the system is not controllable. A more effective approach is to give up (Lipschitz-) continuity at the desired configuration. As shown by M'Closkey and Murray (1997) and Morin and Samson (2000), this allows to design control laws that guarantee a modified form of exponential convergence to the goal.

Most of the aforementioned control designs – both for trajectory tracking and point stabilization – were first developed with reference to the unicycle robot but can be carried out on chained forms, thereby providing an effective extension to other kinematic models, e.g., the car-like robot.

Summary and Future Directions

Wheeled mobile robots are increasingly present in applications. Over the last two decades, significant results have been reached in terms of modeling, planning and control of these systems, and the field is now considered to be well established, at least from an application point of view. Nevertheless, a number of research directions are still open, including the following:

- *Planning and control for non-flat systems*: Relatively harmless wheeled robots (such as a unicycle towing more than one off-hooked trailer) are not flat.
- *Robustness*: The performance of controllers in the presence of disturbances and model perturbations has not received sufficient attention so far.
- *Localization*: Feedback control requires accurate measurements of the configuration variables, which in mobile robots cannot be reliably reconstructed from onboard sensors (odometric data). Integration of exteroceptive sensing is essential to this end.
- *Vision-based control*: As an alternative to localization-based methods, the feedback loop may be closed directly in the image plane, with significant advantages in terms of simplicity and robustness.
- *Multi-robot systems*: The problem is to control the motion of multiple mobile robots in order to perform a cooperative motion task, e.g., formation control.

Cross-References

▶ Differential Geometric Methods in Nonlinear Control
▶ Feedback Linearization of Nonlinear Systems
▶ Feedback Stabilization of Nonlinear Systems
▶ Lie Algebraic Methods in Nonlinear Control
▶ Vehicle Dynamics Control

W

Recommended Reading

For background material on nonlinear controllability, including the necessary concepts of differential geometry, see Sastry (2005). General introductions to mobile robots can be found in Siegwart and Nourbakhsh (2004), Choset et al. (2005), Morin and Samson (2008), and Siciliano et al. (2009). A classification of wheeled mobile robots based on the number, placement, and type of wheels was proposed by Bastin et al. (1996). A detailed extension of some of the planning and control techniques reviewed in this article to the case of car-like kinematics is given in De Luca et al. (1998). A framework for the stabilization of non-flat nonholonomic robots was presented by Oriolo and Vendittelli (2005). Recent work aimed at designing practical universal controllers was carried out by Morin and Samson (2009).

Bibliography

Aicardi M, Casalino G, Bicchi A, Balestrino A (1995) Closed loop steering of unicycle-like vehicles via lyapunov techniques. IEEE Robot Autom Mag 2(1): 27–35

Bastin G, Campion G, D'Andréa-Novel B (1996) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. IEEE Trans Robot Autom 12: 47–62

Brockett RW (1983) Asymptotic stability and feedback stabilization. In: Brockett RW, Millman RS, Sussmann HJ (eds) Differential geometric control theory. Birkhauser, Boston

Canudas de Wit C, Khennouf H, Samson C, Sørdalen OJ (1993) Nonlinear control design for mobile robots. In: Zheng YF (ed) Recent trends in mobile robots. World Scientific, Singapore, pp 121–156

Choset H, Lynch KM, Hutchinson S, Kantor G, Burgard W, Kavraki LE, Thrun S (2005) Principles of robot motion: theory, algorithms, and implementations. MIT, Cambridge

De Luca A, Oriolo G, Samson C (1998) Feedback control of a nonholonomic car-like robot. In: Laumond J-P (ed) Robot motion planning and control. Springer, London, pp 171–253

Fliess M, Lévine J, Martin P, Rouchon P (1995) Flatness and defect of nonlinear systems: introductory theory and examples. Int J Control 61:1327–1361

Lizárraga DA (2004) Obstructions to the existence of universal stabilizers for smooth control systems. Math Control Signals Syst 16:255–277

M'Closkey RT, Murray RM (1997) Exponential stabilization of driftless nonlinear control systems using homogeneous feedback. IEEE Trans Autom Control 42:614–628

Morin P, Samson C (2000) Control of non-linear chained systems: from the Routh-Hurwitz stability criterion to time-varying exponential stabilizers. IEEE Trans Autom Control 45: 141–146

Morin P, Samson C (2008) Motion control of wheeled mobile robots. In: Khatib O, Siciliano B (eds) Handbook of robotics. Springer, New York, pp 799–826

Morin P, Samson C (2009) Control of nonholonomic mobile robots based on the transverse function approach. IEEE Trans Robot 25:1058–1073

Murray RM, Sastry SS (1993) Nonholonomic motion planning: steering using sinusoids. IEEE Trans Autom Control 38:700–716

Neimark JI, Fufaev FA (1972) Dynamics of nonholonomic systems. American Mathematical Society, Providence

Oriolo G, Vendittelli M (2005) A framework for the stabilization of general nonholonomic systems with an application to the plate-ball mechanism. IEEE Trans Robot 21:162–175

Oriolo G, De Luca A, Vendittelli M (2002) WMR control via dynamic feedback linearization: design, implementation and experimental validation. IEEE Trans Control Syst Technol 10: 835–852

Samson C (1993) Time-varying feedback stabilization of car-like wheeled mobile robots. Int J Robot Res 12(1):55–64

Sastry S (2005) Nonlinear systems: analysis, stability and control. Springer, New York

Siciliano B, Sciavicco L, Villani L, Oriolo G (2009) Robotics: modelling, planning and control. Springer, London

Siegwart R, Nourbakhsh IR (2004) Introduction to autonomous mobile robots. MIT, Cambridge

This encyclopedia includes no entries for J, X, Y & Z.